5 SISTEMA N^2OCR

Este capítulo apresentará a proposta de um sistema completo de reconhecimento de caracteres, denominado de sistema N^2OCR - *Neural Network for Optical Character Recognition* [OSO 90a]. O sistema N^2OCR é orientado ao reconhecimento de caracteres impressos de múltiplas fontes, possibilitando o reconhecimento de documentos datilografados, impressos por computador ou textos de livros e revistas. O sistema N^2OCR, conforme será apresentado posteriormente, também prevê uma extensão que possibilitará o reconhecimento de caracteres manuscritos em letras de imprensa (escrita não cursiva).

Baseado nesta proposta de implementação de um sistema completo de reconhecimento de caracteres, foi implementado um protótipo do mesmo, abrangendo as principais funções necessárias à validação das técnicas e algoritmos utilizados junto ao sistema N^2OCR. Deixou-se para um trabalho posterior a implementação das demais funções que compõem o sistema N^2OCR.

As funções, que foram implementadas no protótipo, permitem a utilização deste para o reconhecimento de caracteres, mas de uma forma bem simplificada e sem a flexibilidade necessária para um bom desempenho em sistemas comerciais. Algumas rotinas que permitiriam uma maior flexibilidade ao sistema, assim como rotinas que possibilitassem ao mesmo sua utilização para o processamento de grandes volumes de dados (aumento do automatismo de suas tarefas) não foram implementadas completamente. Desta maneira o protótipo do sistema, como diz o nome, é apenas um protótipo de testes, que permite a avaliação geral da proposta deste trabalho. Logo, o protótipo do N^2OCR *não* é uma ferramenta desenvolvida para fins de uso corrente, e sim para validação da proposta desenvolvida.

A tabela 5.1 apresenta uma lista de funções típicas de sistemas OCR, onde são assinaladas aquelas funções estudadas e previstas no sistema N^2OCR e as funções que foram implementadas no protótipo.

Tabela 5.1 - Funções previstas no sistema N^2OCR e funções implementadas no protótipo

« Funções »	N^2OCR	Protótipo
Aqusição de imagens por: câmera de vídeo ou scanner	●	*[1]
Aquisição de textos através de mesa digitalizadora	X	X
Geração de imagens para teste com distorção controlada	●	●
Armazenamento da imagem: (lê e escr.) formatos CUT, BTM, FIM e IMF	●	●
Armazenamento da imagem: (lê e escr.) formatos TIFF, GIF, PCX e pelo LZW	●	X
Manipulação e visualização de imagens P&B, halftoning, tons de cinza	●	●
Manipulação de imagens coloridas: componentes RGB	X	X
Manipulação de imagens: edição, recorte, junção e justaposição	●	*[1]
Processamento de Imagens: histograma, limiarização, binarização, convolução	●	●
Processamento de Imagens: operações no domínio freqüência, transformadas	X	X
Processamento de Imagens: realce, restauração, filtragem e vetorização	●	X
Automação do processamento com configuração de tarefas (uso off-line)	●	X
Manipulação de imagens de tamanho superior a capacidade de memória	●	*[2]

Tabela 8.1 - Funções previstas no sistema N^2OCR e funções implementadas no protótipo (*Continuação*)

« Funções »	N^2OCR	Protótipo
Reconhecimento de caracteres isolados de múltiplas fontes impresso ⟶	●	●
manuscrito (letra de imprensa) ⟶	●	*[3]
Reconhecimento de caracteres em letra cursiva ou com escrita on-line	X	X
Reconhecimento de caracteres em tempo real (uso de paralelismo)	●	X
Reconhecimento de caracteres que se tocam (touching characters)	●	*[2]
Conjunto de caracteres sem limitações OCR ANSI, alfabeto latino, grego, etc	●	●
Resultado do reconhecimento: ASCII, BRASCII, EBCDIC, outras codificações	●	●
Sem restrições quanto a escrita: tamanho, formulários especiais, estilo	●	●
Pré-processamento: separação de textos e gráficos	●	*[1]
Pré-processamento: isolar caracteres espaçamento variável (por contorno)→	●	●
espaçamento fixo (pelo tamanho) ⟶	●	X
Pré-processamento: ajuste de escala e posição dos caracteres	●	●
Pré-processamento: extração de novos atributos e recodificação dos padrões	●	X
Pós-processamento: geração de lista de candidatos para análise de contexto	●	●
Pós-processamento: reconhecimento com uso de contexto na correçao de erros	●	X
Capacidade de avaliação do reconhecimento: grau de acerto, erro, não id.	●	●
Sist. de reconhecimento com capacidada de aprendizado tutoriado interativo	●	●

Tabela 5.1 - Funções previstas no sistema N^2OCR e funções implementadas no protótipo (*Continuação*)

« Funções »	N^2OCR	Protótipo
Sistema de reconhecimento com capacidade de adaptação do aprendizado	●	●
Sistema de reconhecimento com capacidade de emprego de paralelismo no processamento	●	X
Rede neural: reconhecimento do XOR, expansível, aprendizado supervisionado por correção de erros	●	●

Notação:

● - Função prevista (N^2OCR) ou implementada (protótipo)

X - Função não prevista e não implementada

$*^1$- Possível através da utilização de programa externo (uso de ferramenta que não faz parte do sistema)

$*^2$- Possível de ser realizado, mas de maneira "restrita"

$*^3$- Não foi testada/avaliada esta possibilidade de uso

O problema da falta de flexibilidade encontra-se, por exemplo, no algoritmo de segmentação de caracteres, que possui implementado no protótipo apenas a segmentação de caracteres com espaçamento variável por varredura do contorno. Este algoritmo funciona bem com textos datilografados ou impressos (se for no computador, através de impressora laser) de espaçamento variável. O mesmo algoritmo não tem um bom desempenho com textos impressos através de uma impressora matricial, e para este caso é necessária a implementação de um novo algoritmo de segmentação para caracteres com espaçamento fixo.

As seções seguintes deste capítulo descreverão o sistema N^2OCR em termos de suas características gerais, e a implementação de seu protótipo.

5.1 Organização Geral do Sistema

O sistema N^2OCR é composto por vários módulos que implementam as funções da aquisição das imagens até a saída com os caracteres reconhecidos. Os principais módulos que compõem o sistema são:

- Módulo de aquisição e geração de imagens;
- Módulo de tratamento das imagens;
- Módulo de preparação dos dados para a rede neural e
- Módulo simulador da rede neural.

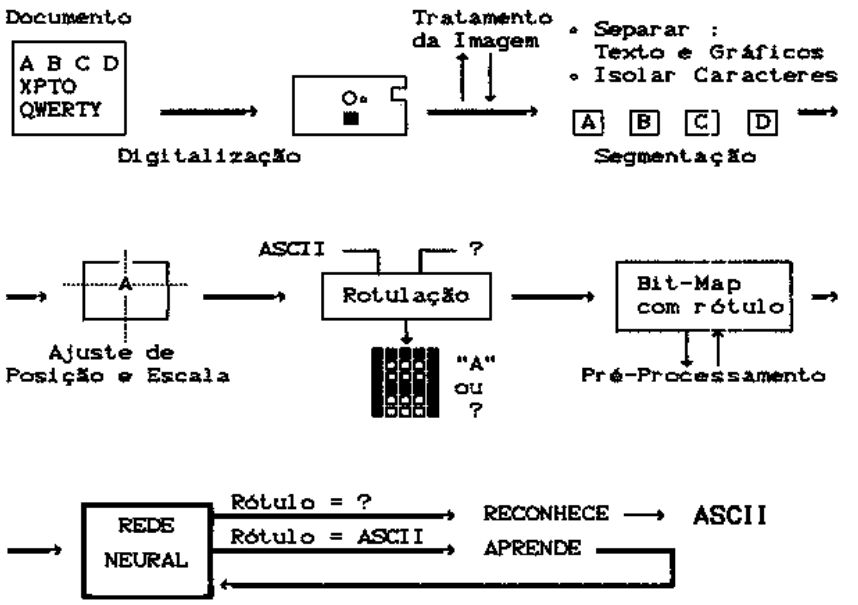


Figura 5.1 - Esquema de funcionamendo do sistema N^2OCR

A figura 5.1 apresenta um esquema geral da organização do sistema N^2OCR, onde estão representadas as diversas etapas de processamento realizada pelo sistema através de cada um destes módulos que o compõem. Nas próximas seções serão abordados em detalhes cada módulo que forma o sistema. Na figura 5.2 são apresentados os diferentes módulos que foram implementados no protótipo do sistema N^2OCR, juntamente com as funções disponíveis de cada módulo e a interação entre estes.

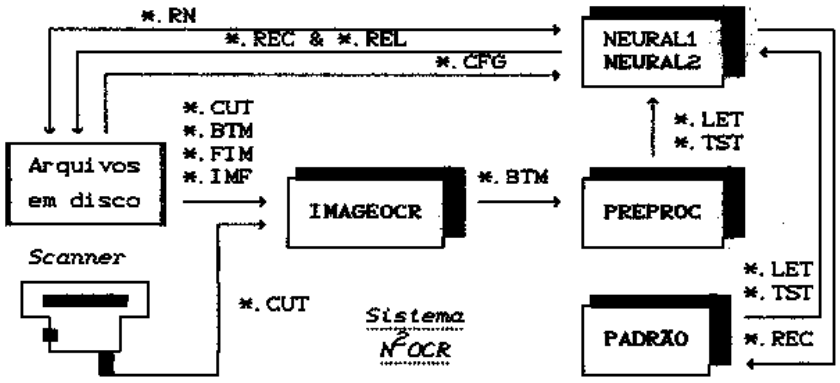


Figura 5.2 - Protótipo do sistema N^2OCR

Os módulos que compõem o protótipo do sistema N^2OCR e que aparecem representados na figura 5.2 são os seguintes:

- Padrão : (padrão.exe)
  - Definição de parâmetros de geração e distorção de imagens textuais;
  - Geração de padrões: com ou sem distorção, para aprendizado ou reconhecimento (gera rótulos);
  - Visualização do arquivo de padrões;
  - Avaliação do reconhecimento.

- Imageocr : (imageocr.exe)

  - Visualização de imagens (P&B até 256 níveis);
  - Conversão de formatos de arquivos de imagens;
  - Recorte de imagens;
  - Geração de histograma;
  - Alteração do histograma: compressão e expansão da escala de cinza, limiarização e binarização;
  - Realização de operações de convolução (3x3).

- Preproc : (preproc.exe)

  - Visualização de imagens e arquivo de padrões;
  - Separação dos caracteres;
  - Rotulação dos padrões;
  - Ajuste de posição e escala;
  - Extração de atributos (não implementado).

- Neural2 : (neural2.exe)

  - Rede neural → modelo ADAnLIeNE;
  - Configuração do simulador (arquivos *.CFG);
  - Aprendizado de padrões (arquivos *.LET);
  - Reconhecimento de padrões (arquivos *.TST);
  - Salvamento e recuperação de redes neurais já treinadas (arquivos com pesos da rede *.RN);
  - Geração de relatório de acompanhamento da simulação (arquivos *.REL);
  - Geração do texto reconhecido, incluindo ou não os candidatos alternativos para pós-processamento (arquivos *.REC).

- Neural1 : (neural1.exe)

  - Rede neural → modelo ADALINE
  - Similar ao neural2 (1ª versão do simulador, sem a capacidade de expandir a rede;

5.2 Aquisição de Imagens

A aquisição de imagens pode ser realizada através de diferentes dispositivos, como câmeras de vídeo e *scanners*, conforme foi discutido no capítulo 2. O protótipo do sistema N^2OCR permite a utilização de imagens obtidas através de algum destes dispositivos, podendo manipular apenas imagens monocromáticas, mas com um número variável de tonalidades. O módulo de aquisição de imagens baseia-se na utilização de um *handy scanner*, o DFI HS-3000 [FAL 88]. A captura e manipulação de imagens (recorte, junção, edição das imagens) é feita utilizando-se um programa de editoração eletrônica, o *HaloDPE*, fornecido pelo fabricante juntamente com o equipamento.

Devido ao uso de ferramentas prontas, a forma de acesso às imagens é através de arquivos gravados por estes programas. Foi desenvolvida uma série de programas para conversão de arquivos para diferentes formatos, sendo alguns destes utilizados no sistema implementado. O formato dos arquivos utilizados junto ao sistema N^2OCR e ao HaloDPE estão descritos no anexo A.1 e A.2.

Para tornar o sistema N^2OCR viável, é necessário que se desenvolva uma ferramenta própria para a captura de imagens, pois a manipulação através de arquivos gravados em disco torna o processo muito demorado. Portanto, um módulo, não implementado no protótipo, a ser incorporado ao sistema é um módulo que realizasse as tarefas de aquisição das imagens e sua manipulação diretamente através do sistema N^2OCR, sem a necessidade de entrar e sair de aplicações, e onde os dados seriam manipulados preferencialmente sempre na memória, agilizando o processo de reconhecimento.

O sistema N^2OCR, além de permitir a captura direta de imagens, também deve permitir o acesso a arquivos de imagens gerados por outros sistemas. Desta forma o N^2OCR pode operar acoplado a um sistema específico de aquisição de imagens, mas também pode aceitar o processamento de imagens obtidas por outros dispositivos.

Figura 5.3 - Tela de ajuda do módulo gerador de imagens

Figura 5.4 - Configuração do módulo gerador de imagens

A obtenção de imagens para uso junto ao sistema N^2OCR foi implementada de duas formas: através do acesso a arquivos de imagens gerados pelo *scanner* HS-3000, ou através de um programa que permite a geração de um *bit-map*, simulando a captura de um texto através de um dispositivo como os *scanners*.

O módulo gerador de imagens (programa padrão.exe - fig. 5.3) é um programa que realiza a função inversa de um sistema de reconhecimento de caracteres, ou seja, dado um texto qualquer, o programa gerará um *bit-map* com a imagem do texto fornecido. Foi implementado um gerador de imagens textuais que permite a configuração de alguns parâmetros, apresentados na figura 5.4, tais como:

- Seleção de diferentes fontes (tipos de caracteres);
- Seleção do tamanho dos *bit-maps* gerados para cada caractere e
- Geração simulada de perturbações no texto: inversão aleatória de pontos (figs. 5.5, 5.6 e 5.7), rotação (fig. 5.8) ou translação dos caracteres gerados.

O objetivo deste programa de geração de imagens textuais foi o de possibilitar uma análise do sistema de reconhecimento, independentemente da possibilidade de se utilizar ou não um sistema de aquisição real de imagens. Como o dispositivo de aquisição de imagens só se tornou disponível em uma fase mais adiantada da implementação do sistema, tal programa foi de destacada importância na implementação do N^2OCR. Outra característica muito importante deste programa de geração de imagens é a possibilidade de um controle exato dos efeitos de distorção sobre o texto, permitindo desta forma uma melhor análise e acompanhamento do desenvolvimento do sistema de reconhecimento de caracteres, como demonstram as figuras 5.5 a 5.8.

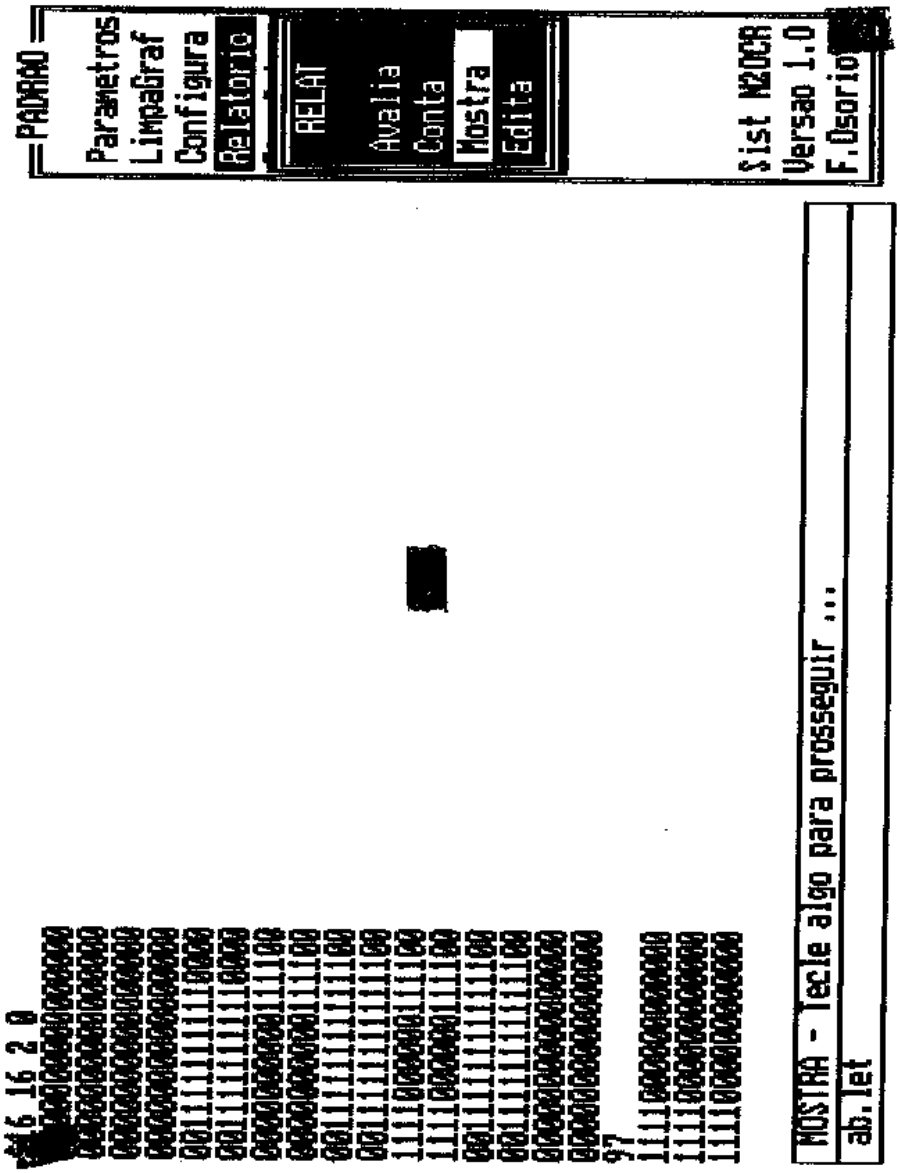


Figura 5.5 - Padrão do caractere 'a' gerado sem distorção

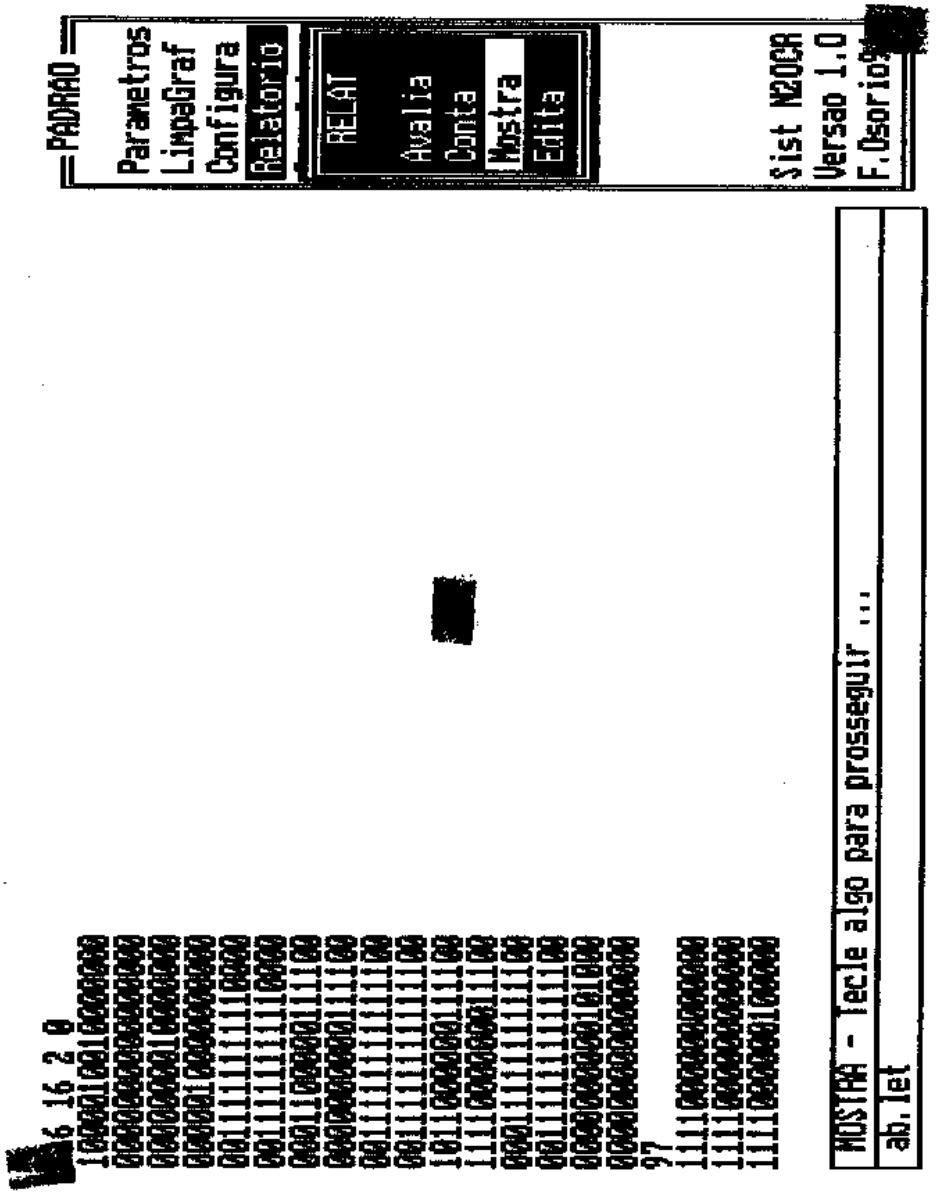


Figura 5.6 - Padrão do caractere 'a' gerado com a inversão aleatória de 5% dos pixels

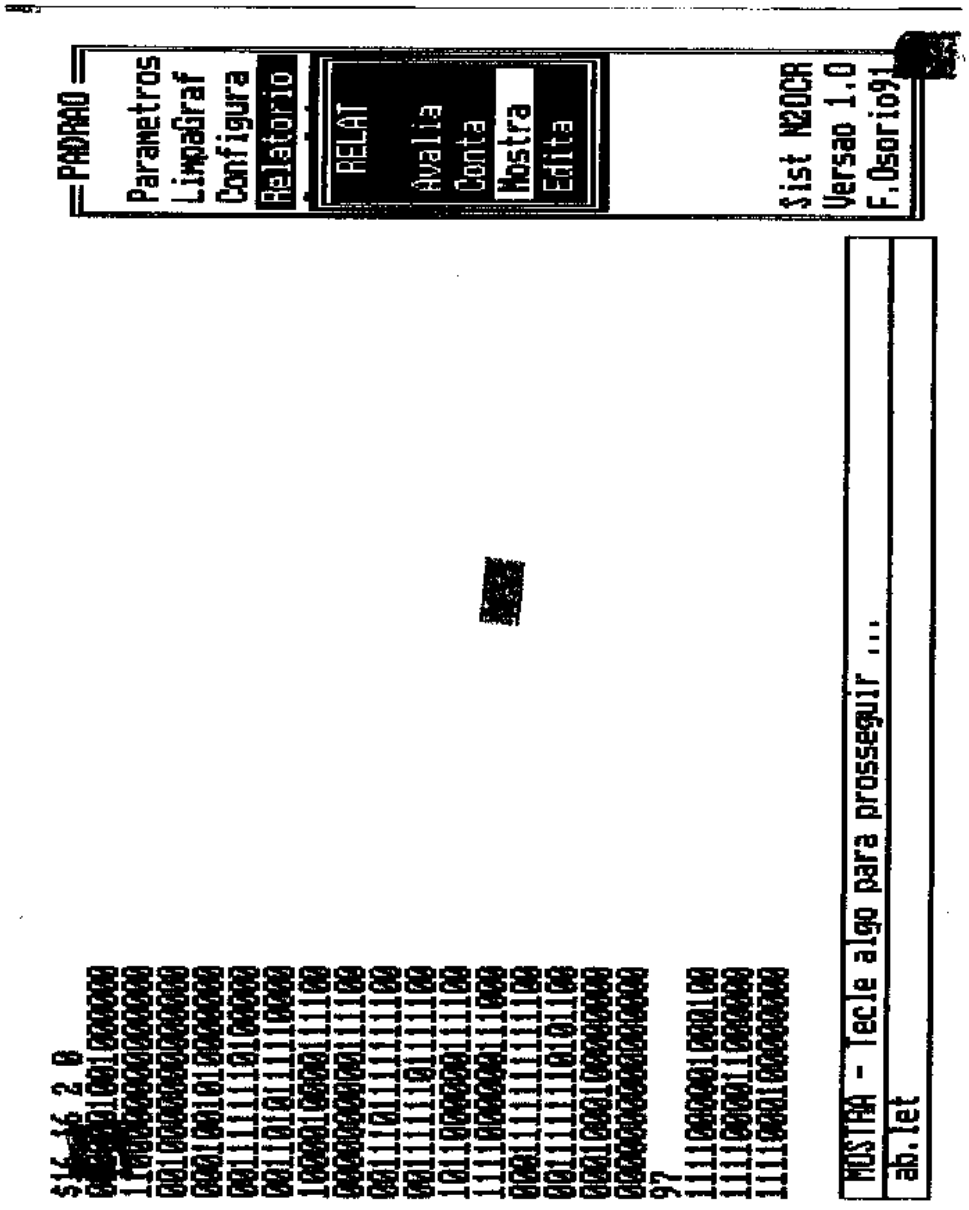


Figura 5.7 - Padrão do caractere 'a' gerado com a inversão aleatória de 10% dos pixels

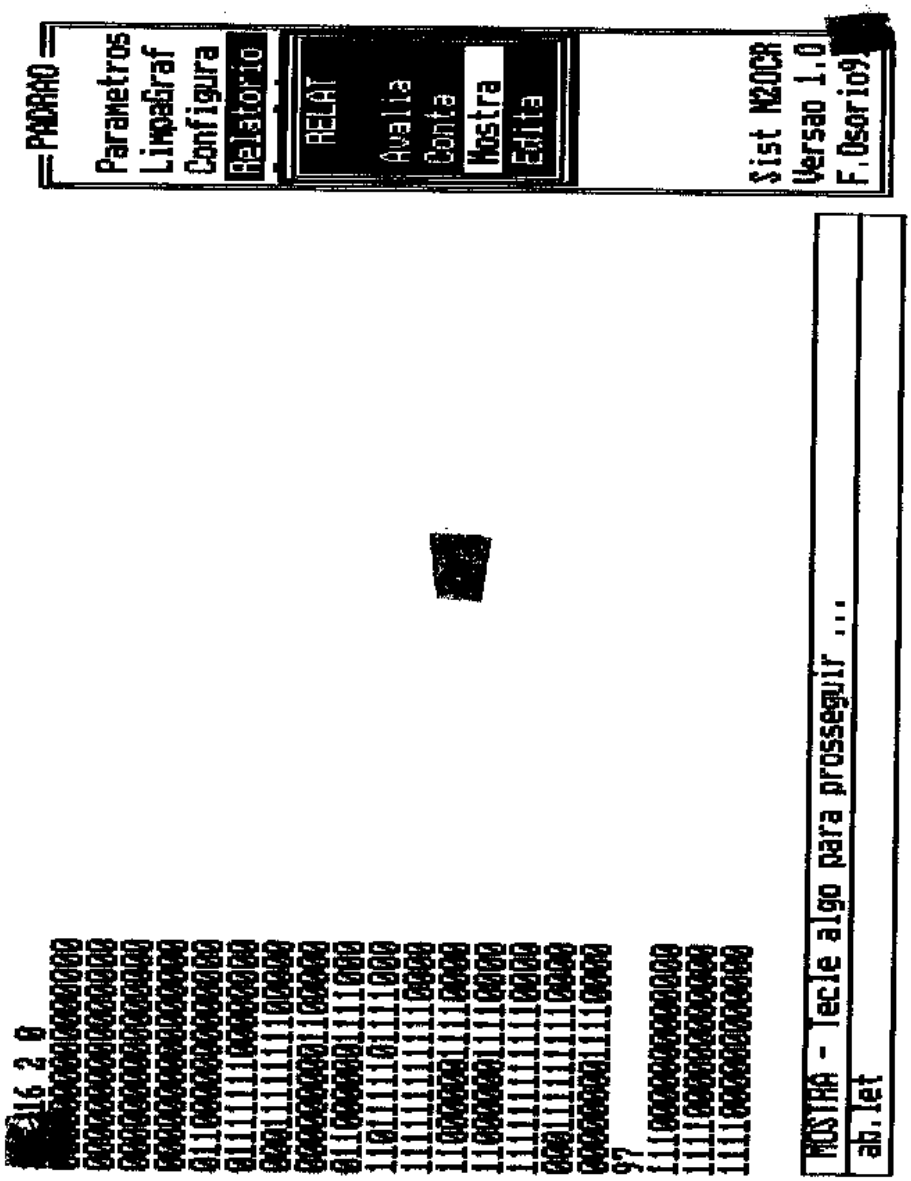


Figura 5.8 - Padrão do caractere 'a' gerado com rotação de 10 graus

5.3 Tratamento da Imagem

O sistema N^2OCR possui um módulo dedicado ao processamento das imagens de documentos digitalizados (programa imageocr.exe - fig. 5.9). Este módulo implementa algumas das funções descritas no capítulo 2, tais como:

- Realização de conversões de diferentes tipos de formatos de armazenamento de imagens em arquivos (fig.5.10);
- Geração de uma imagem com uma escala de tonalidades reais, a partir de uma imagem com *halftoning* (desfazer o *halftoning*);
- Manipulação da imagem, possibilitando o recorte de uma área específica para "salvamento" em disco;
- Visualização de imagens monocromáticas de 2 até 256 níveis de intensidade (figs. 5.11 e 5.12);
- Geração do histograma de uma imagem (fig. 5.13);
- Alteração do histograma da imagem: limiarização, binarização (fig. 5.14) e equalização, compressão ou expansão da escala de cinza;
- Realização de operações de convolução para detecção de pontos, linhas horizontais, verticais, diagonais e bordas (fig. 5.15), e
- Aplicação na imagem de um operador de convolução definido pelo usuário.

Todas as operações citadas acima, descritas em detalhes no capítulo 2, foram implementadas no protótipo do sistema N^2OCR, facilitando assim a manipulação das imagens digitalizadas. Algumas operações de tratatamento de imagens poderiam ainda ser adicionadas a este módulo do sistema, incluindo funções como: aplicação de filtros (*smoothing*, *sharpening*, etc) e realização de transformadas. Podem também ser estendidos os formatos de arquivos manipulados pelo sistema, que se restringem atualmente aos formatos: CUT (usado no HaloDPE), BTM e FIM (formatos internos do sistema N^2OCR) e o formato IMF (padrão definido no CPGCC/UFRGS) - Anexo A.1.

Figura 5.9 - Tela de ajuda do módulo de tratamento de imagens

Figura 5.10 - Conversão dos formatos de arquivos

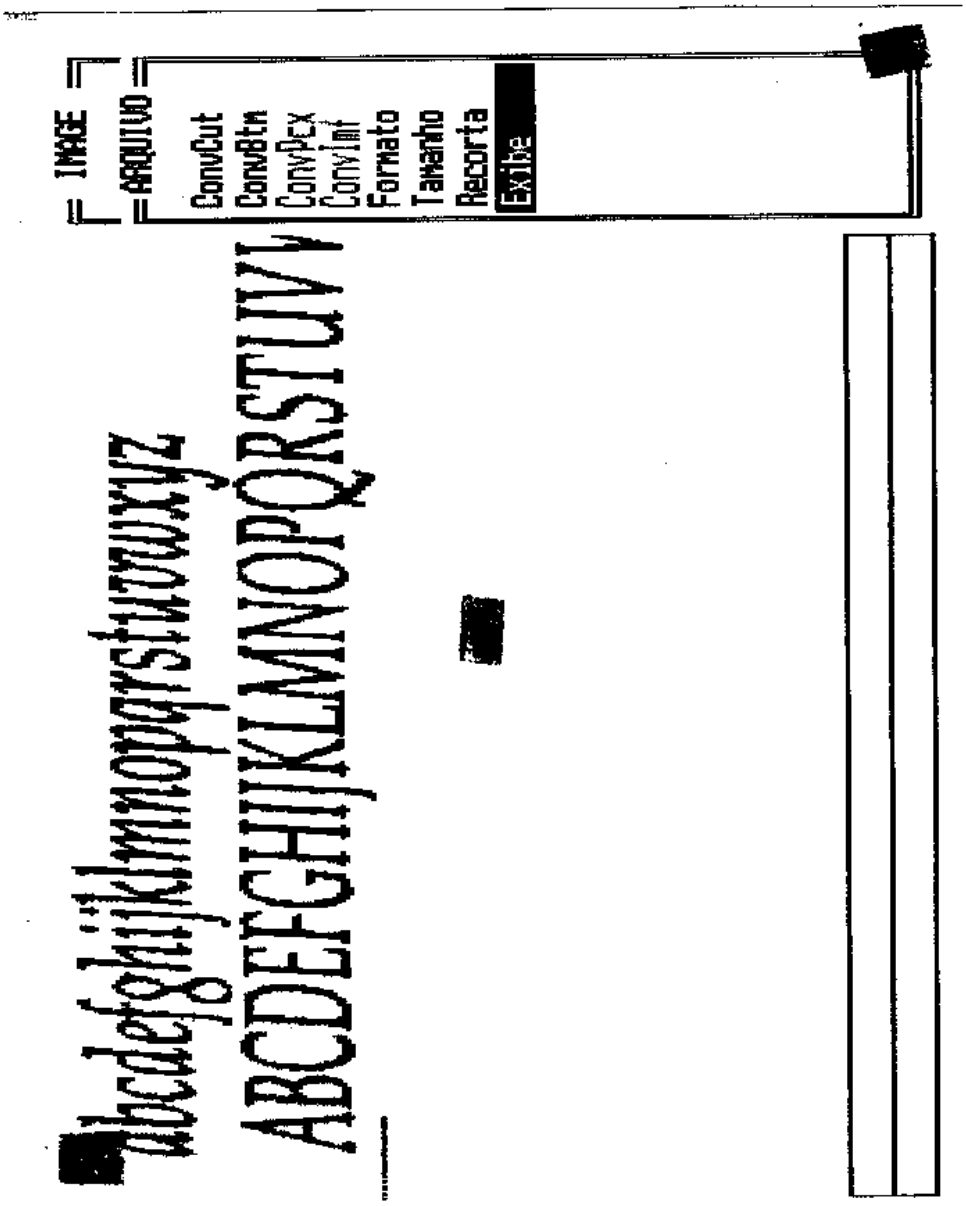


Figura 5.11 - Visualização de uma imagem de um texto

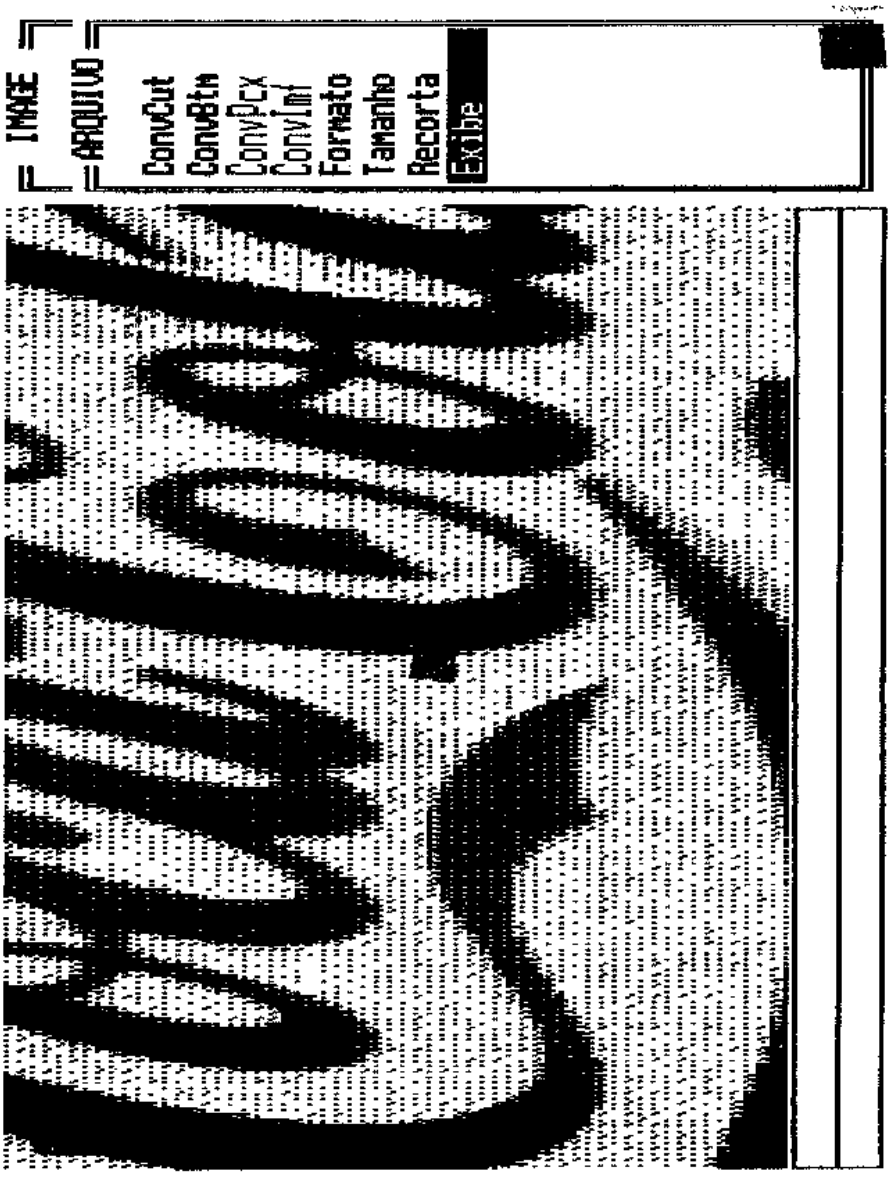


Figura 6.12 - Visualização de uma imagem com halftoning

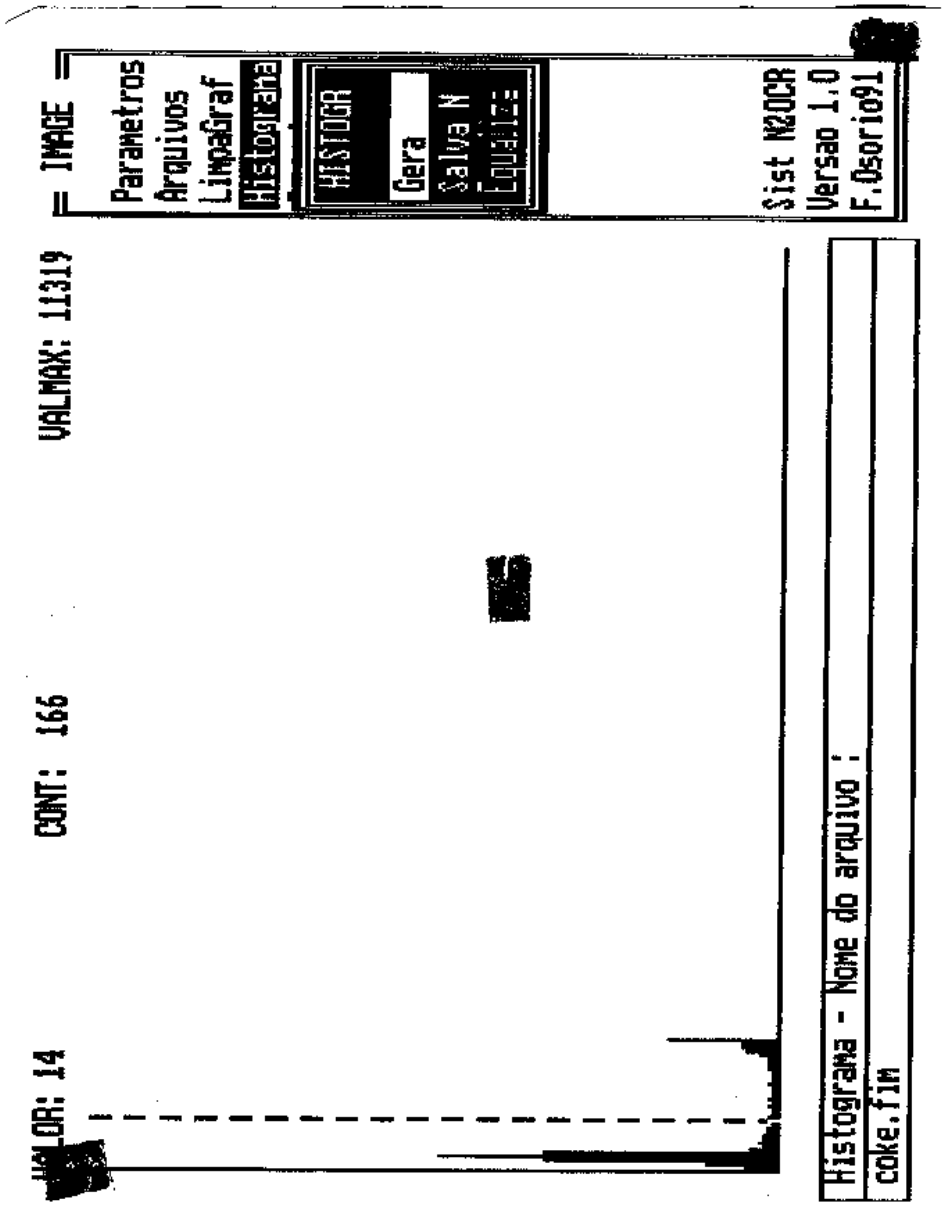


Figura 5.13 - Histograma da imagem da fig. 5.12 com half-toning desfeito

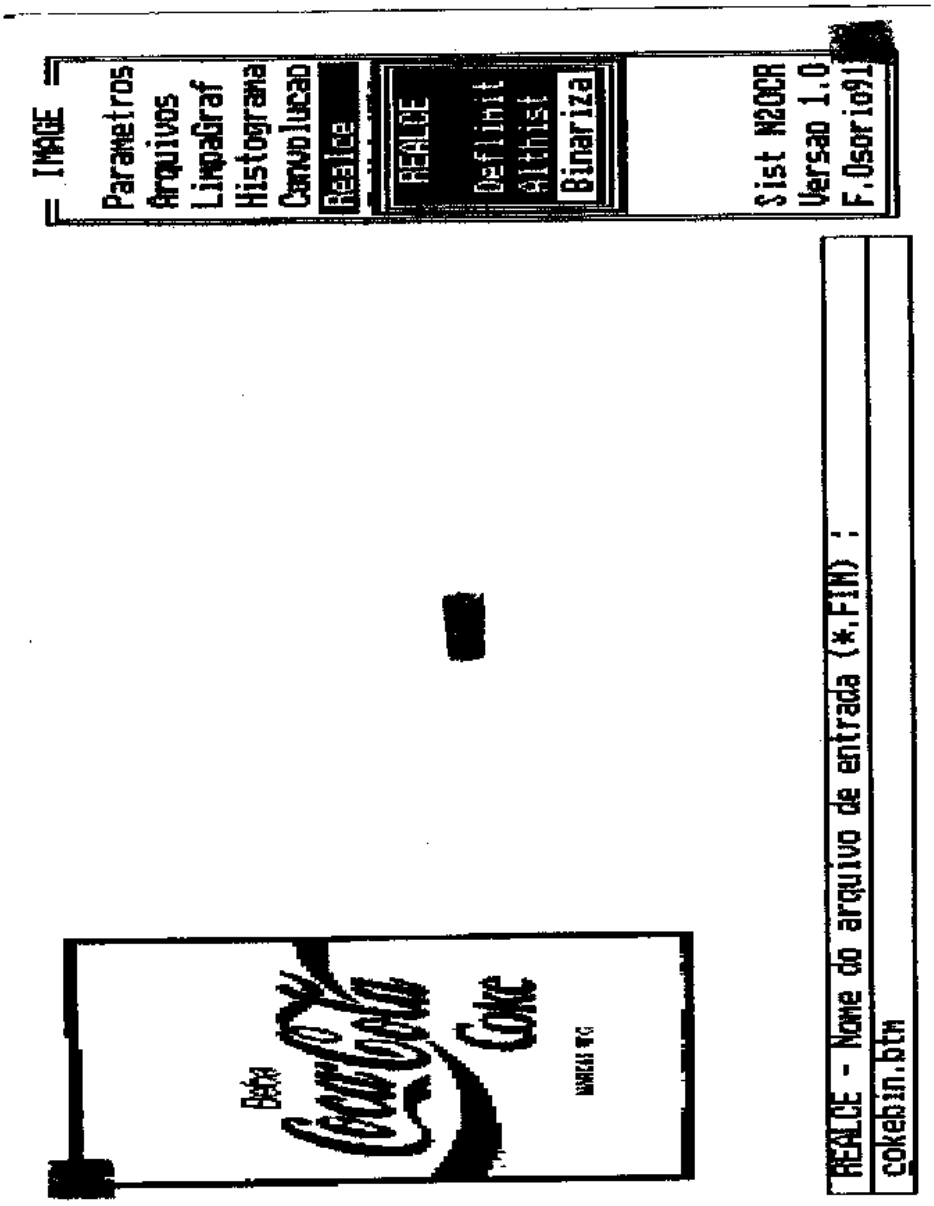


Figura 5.14 - Binarização da imagem cujo histograma aparece na fig. 5.13

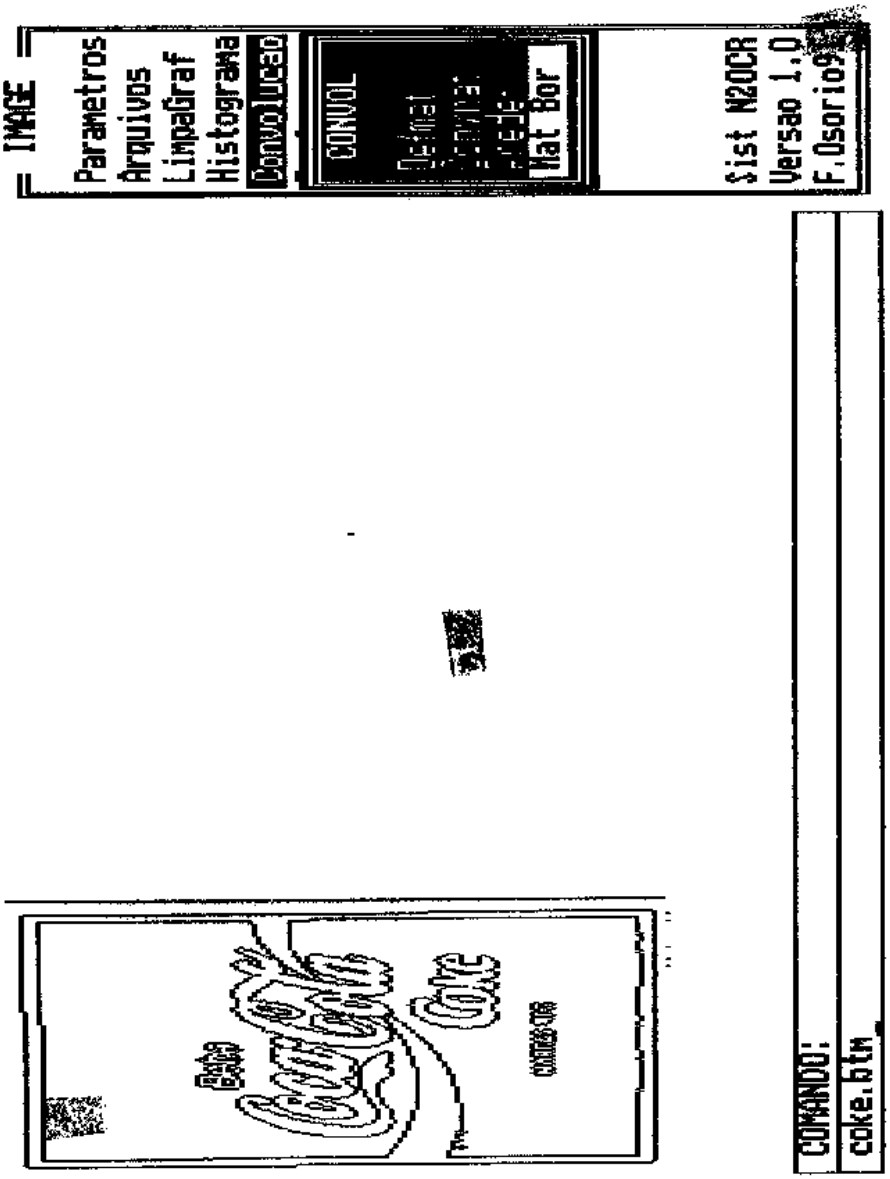


Figura 5.15 - Convolução para detecção de bordas na imagem da fig. 5.14

O módulo de tratamento de imagens consiste de um programa de processamento de imagens essencialmente interativo. Através da visualização da imagem, do seu histograma e do resultado da aplicação de filtros e operações de convolução, o usuário poderá analisar melhor as imagens e optar pelo tipo de processamento que seja mais adequado para um dado documento. Através deste processo interativo de visualização e de processamento da imagem, pode-se, por exemplo, estabelecer o limiar ótimo para a binarização de uma imagem, ou ajustar de uma maneira mais adequada o contraste.

O módulo de tratamento de imagens deve permitir que sejam determinados fatores e procedimentos ideais para um dado tipo de documento a ser digitalizado. Após, este sistema deve permitir a aplicação automática destes procedimentos pré-fixados para toda uma série de documentos a serem processados. Desta forma o usuário pode fixar interativamente um procedimento de tratamento para um certo tipo de documento e posteriormente tornar esta tarefa de processamento automatizada. O protótipo do sistema N^2OCR ainda não provê este tipo de facilidades, onde o tratamento da imagem, quando necessário, terá que ser todo realizado com a supervisão/interação com o usuário.

5.4 Preparação dos Dados para a Rede Neural

Depois da digitalização e do tratamento global de um documento, ele é passado para o módulo de preparação dos dados para a rede neural (programa preproc.exe - fig. 5.16). Este módulo extrai os caracteres, um a um, de forma que possam ser identificados pela rede neural. Neste módulo, são implementadas as funções de segmentação e ajuste de posição e escala dos caracteres, além de possibilitar também uma análise visual das imagens por ele manipuladas (fig. 5.17 e 5.18).

Figura 5.16 - Tela de ajuda do módulo de preparação dos dados

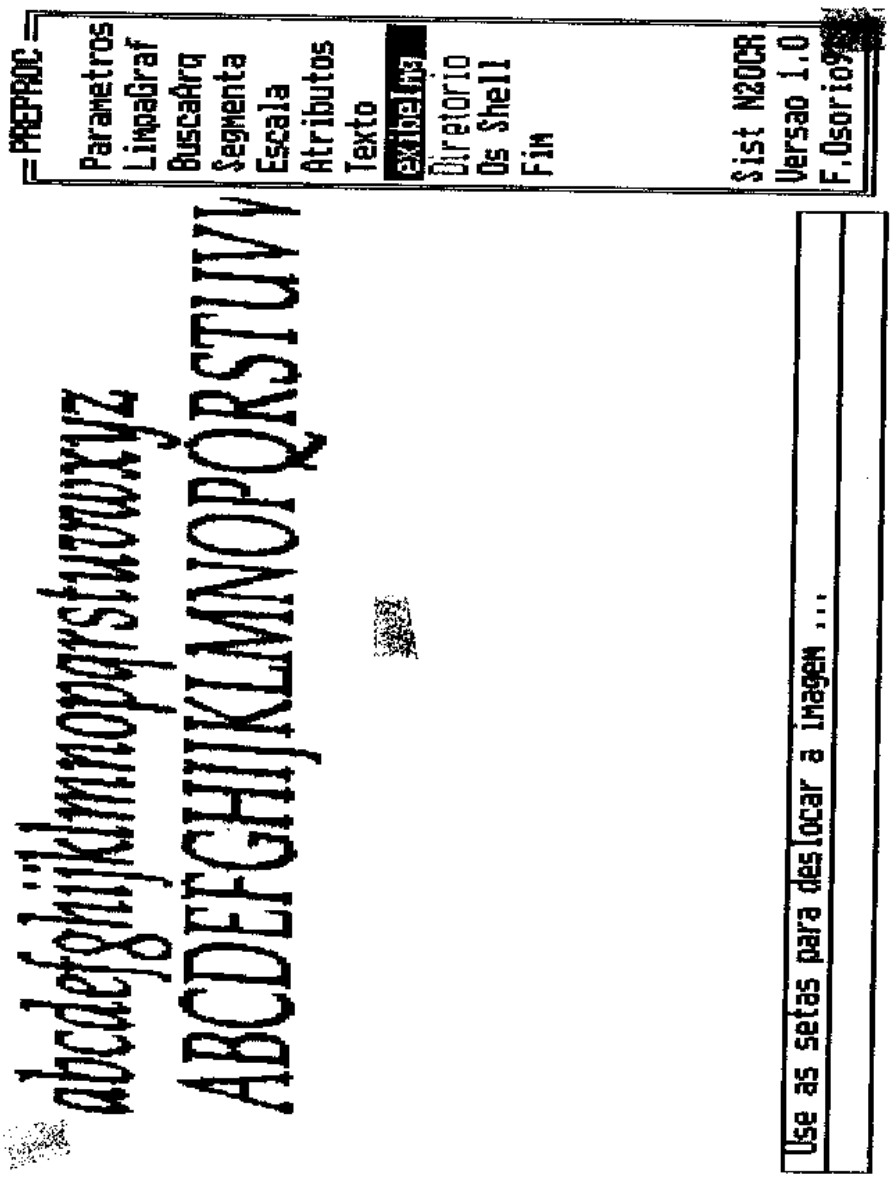


Figura 5.17 - Imagem exibida pelo módulo de preparação dos dados

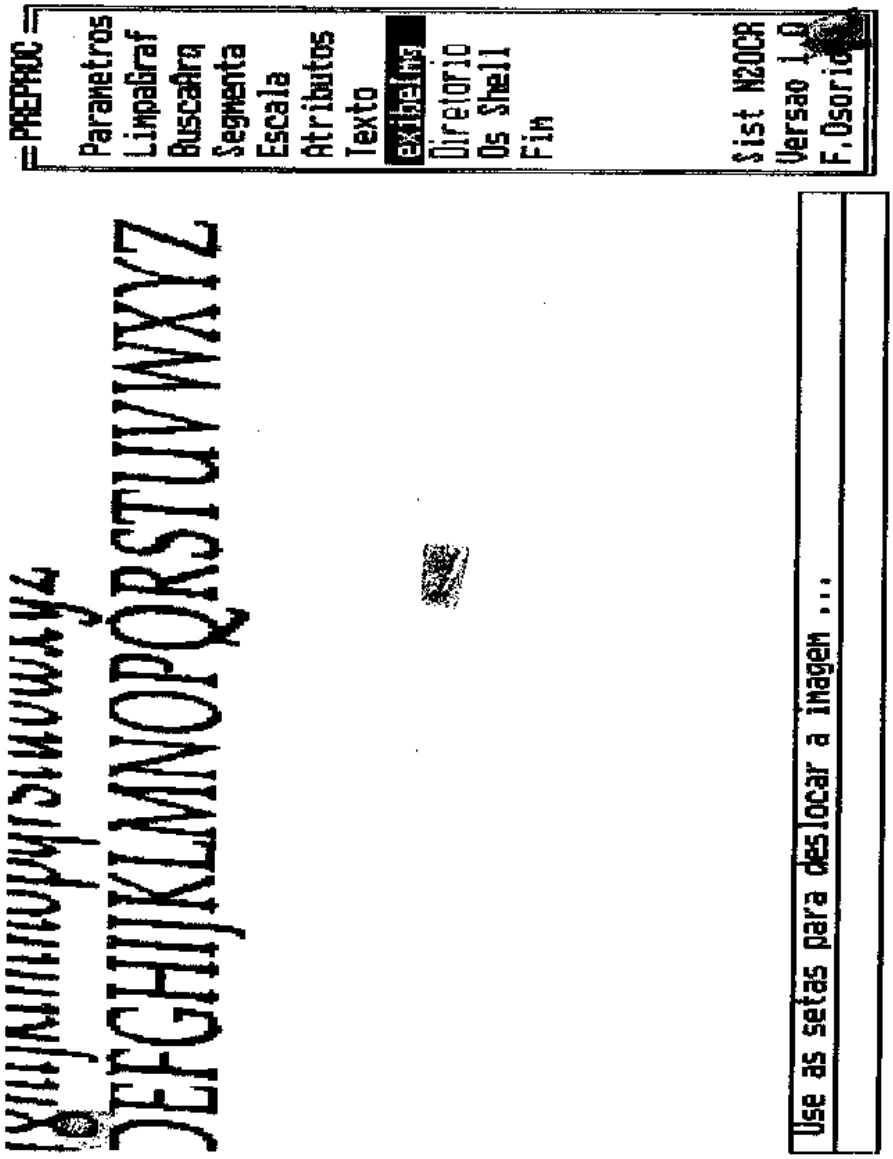


Figura 5.18 - Exibição de outra área da imagem da fig. 5.17 (deslocamento interativo)

Junto ao módulo de preparação dos dados, poderá ser incluida uma etapa de pré-processamento dos padrões a serem reconhecidos. Este assunto será abordado em maiores detalhes na seção 5.8.

5.4.1 Segmentação

O procedimento de segmentação é dividido em duas etapas: a separação dos textos e gráficos, e a separação dos caracteres entre si.

A separação entre textos e gráficos consiste da aplicação do algoritmo descrito na seção 3.4.1.1 (algoritmo RLSA) resultando uma imagem com os textos apenas, onde todos os gráficos são removidos de forma a não prejudicar o processo de reconhecimento. Cada bloco de texto isolado através do algoritmo RLSA será posteriormente submetido a um algoritmo de separação dos caracteres. Também há a opção de ser realizada a separação de gráficos e textos de forma interativa (o usuário edita a imagem do documento, isolando os textos dos gráficos), mas tal processo é lento e dificulta bastante a utilização prática do sistema de reconhecimento.

A segmentação de caracteres consiste da aplicação de um algoritmo para a separação de caracteres com espaçamento variável, ou através da aplicação de um algoritmo de separação de caracteres com espaçamento fixo. Através de uma configuração prévia é feita a seleção de qual dos dois algoritmos será utilizado na segmentação dos caracteres, definindo-se também os outros parâmetros necessários para a realização destas funções (fig 5.19). O algoritmo de segmentação para caracteres com espaçamento variável utilizado é aquele descrito na seção 3.4.1.2, sendo baseado na continuidade das linhas que compõem cada caractere. O algoritmo para a separação de caracteres com espaçamento fixo, também descrito na seção 3.4.1.2, quebra a imagem em pequenos blocos com dimensões aproximadas de M x N pixels, definida pelo usuário.

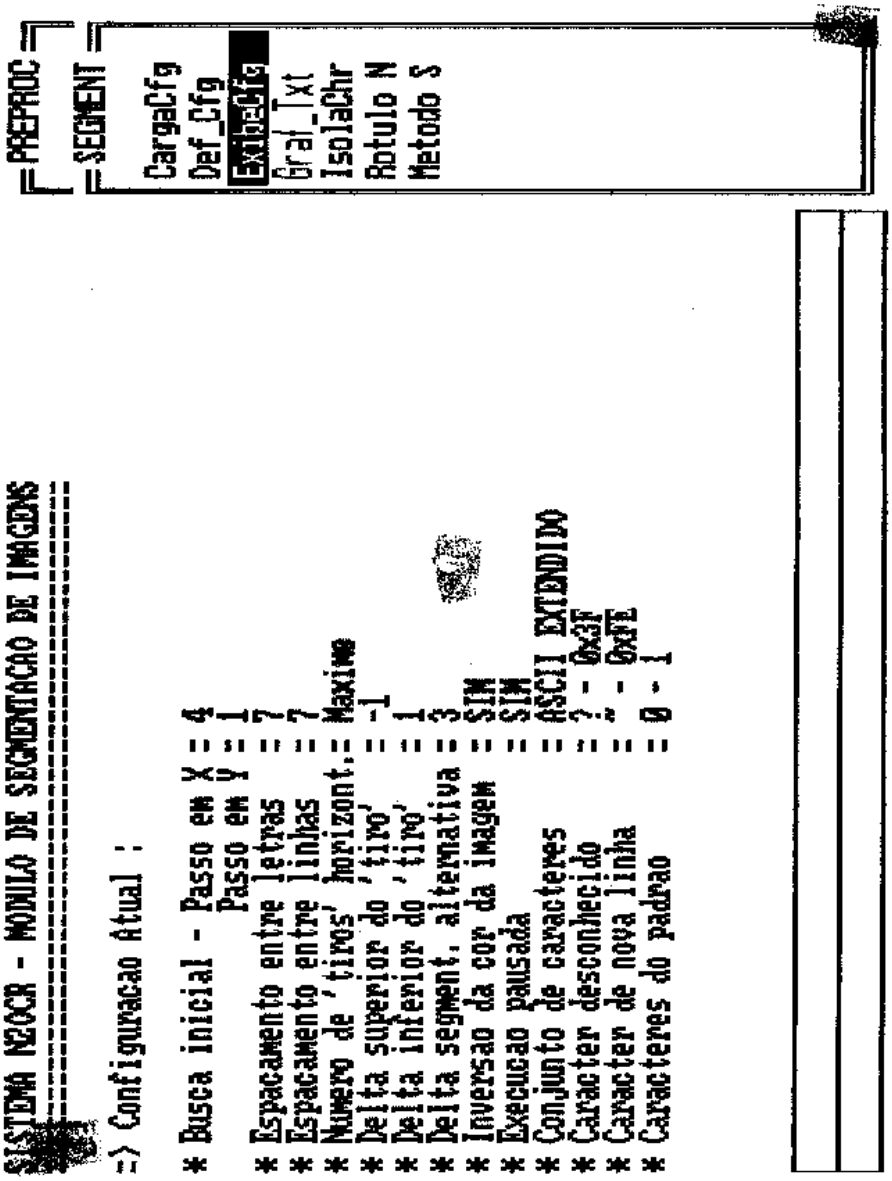


Figura 5.19 - Parâmetros de configuração da segmentação

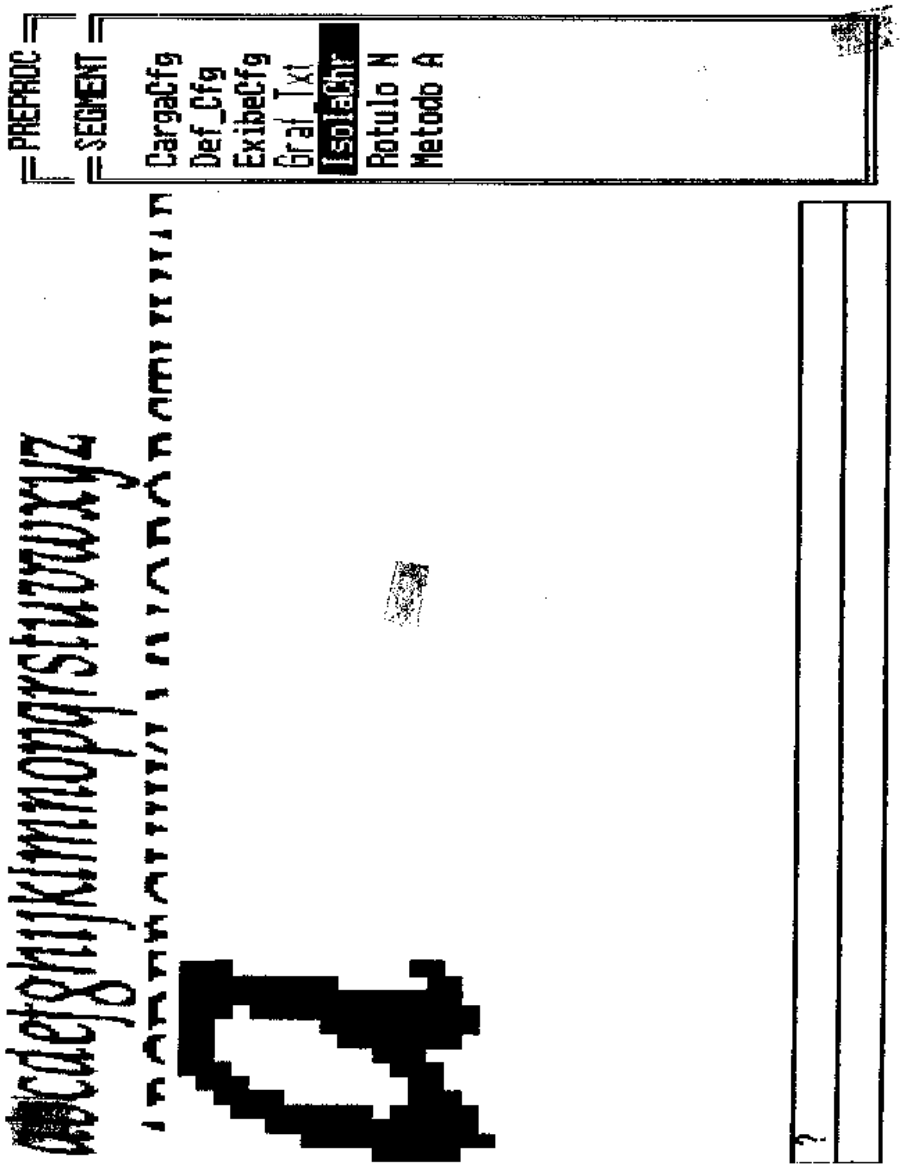


Figura 5.20 - Primeiro caractere isolado da imagem

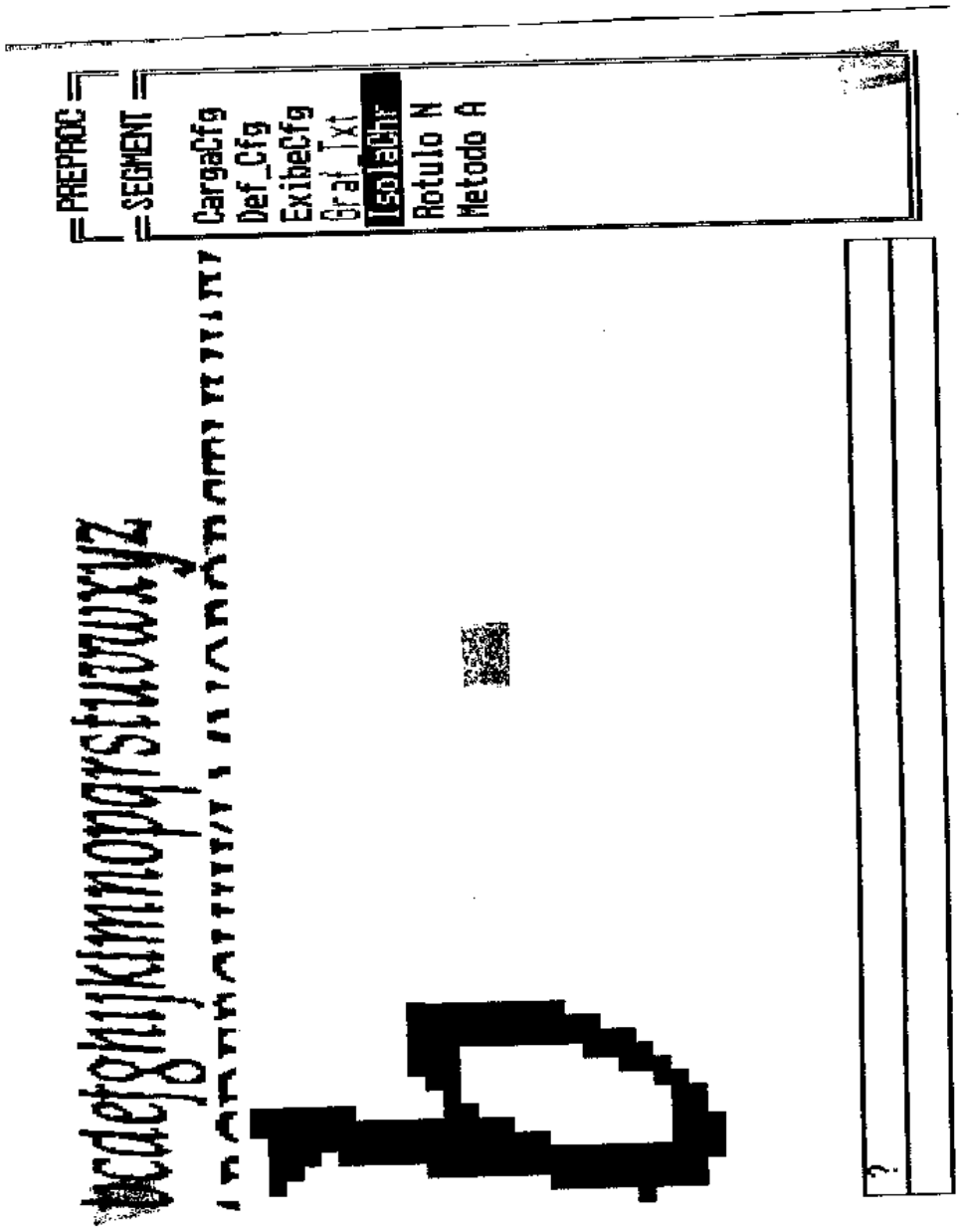


Figura 5.21 - Segundo caractere isolado da imagem

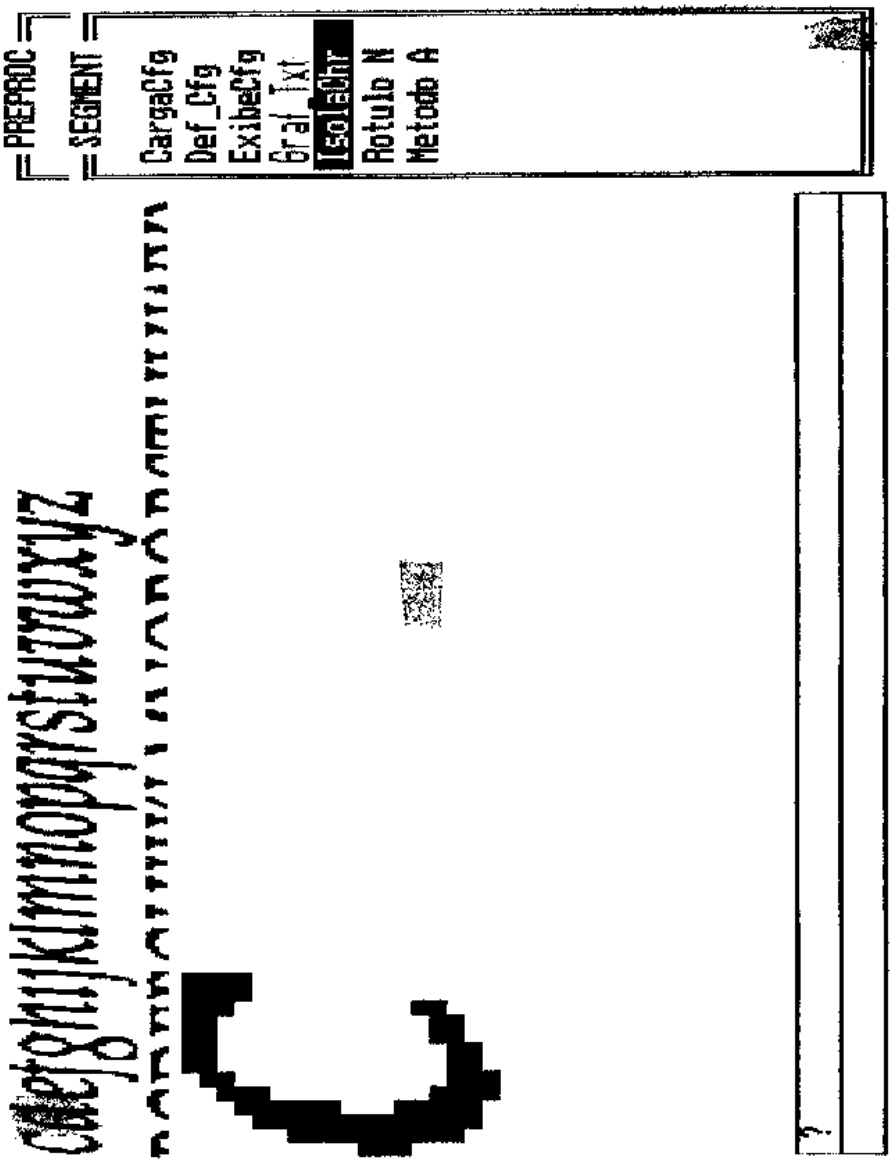


Figura 5.22 - Terceiro caractere isolado da imagem

O protótipo do sistema N^2OCR implementa apenas o algoritmo de separação de caracteres com espaçamento variável. Foi implementado somente este algoritmo porque pode ser utilizado na maioria dos casos de reconhecimento de caracteres em textos impressos. A forma como foi implementado o algoritmo de segmentação permite que o usuário tenha uma realimentação do andamento do procedimento de segmentação, ajudando ao usuário no estabelecimento dos parâmetros de configuração da função e, conseqüentemente, permitindo a obtenção de melhores resultados. As figuras 5.17 e 5.18 apresentam os recursos implementados no módulo de preparação dos dados que facilitam a visualização da imagem, e as figuras 5.20, 5.21 e 5.22 apresentam o desenvolvimento passo a passo da segmentação de caracteres.

O processo de segmentação de caracteres resulta em um conjunto de padrões, com tamanhos diferentes, representando os caracteres isolados. O prótotipo do sistema N^2OCR gera em disco um arquivo que irá conter os caracteres que foram isolados, onde a descrição deste arquivo é apresentada no Anexo A.2.

A separação de caracteres com espaçamento fixo não está disponível na atual versão do sistema e a separação entre textos e gráficos deve ser realizada de forma interativa, ou seja, o usuário deve editar as imagens através de algum programa, como o HaloDPE, eliminando manualmente os gráficos encontrados junto ao texto. No entanto, a falta destes dois algoritmo na implementação do protótipo do sistema N^2OCR não prejudica a sua funcionalidade para fins de teste e validação do mesmo, uma vez que existem métodos alternativos que permitem a sua utilização mesmo sem tais algoritmos implementados.

5.4.2 Rotulação

Uma outra fase da preparação dos dados para a rede neural é a rotulação dos padrões já segmentados.

Esta etapa consiste em adicionar aos padrões um rótulo com a sua identificação, onde este rótulo será utilizado pela rede neural na fase de aprendizado dos padrões. O processo de rotulação dos padrões foi implementado no protótipo de forma a ocorrer juntamente com o procedimento de separação dos caracteres.

A rede neural adotada pelo sistema N²OCR é do tipo de rede que recebe um aprendizado supervisionado, ou seja, o usuário precisa apresentar à rede os padrões e a resposta correta que deve ser obtida como saída desta. Para que seja possível este tipo de procedimento, é necessário que, na fase de treinamento, os padrões sigam acompanhados de uma identificação (resposta correta do reconhecimento), permitindo o aprendizado. A identificação ou rótulo nada mais é do que o próprio código ASCII do padrão em questão, conforme demonstra a figura 5.23.

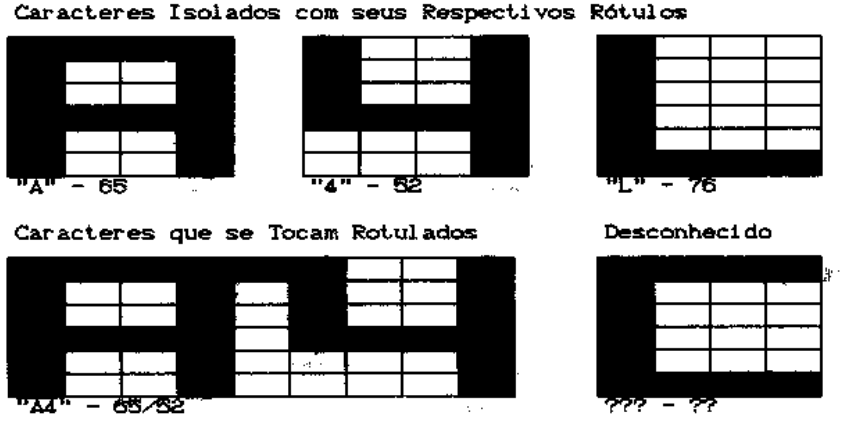


Figura 5.23 - Rotulação de Caracteres

O processo de colocação de rótulos é um processo tipicamente interativo, onde, após serem segmentados, os caracteres são apresentados na tela, um a um, de forma que o usuário possa então adicionar os rótulos a cada um destes (fig. 5.20, 5.21 e 5.22).

O usuário pode optar pela realização de uma segmentação dos caracteres, onde a cada caractere isolado será pedido de maneira interativa a indicação do seu código de identificação, ou, no caso de arquivos para reconhecimento, onde os caracteres não são conhecidos, é possivel se configurar o sistema para a realização da segmentação sem a necessidade da intervenção do usuário (fig. 5.24). Neste último caso a adição do rótulo que identifica justamente que os códigos de identificação dos caracteres não são conhecidos se dará de forma automática.

A adição de rótulos consiste em observar o caractere isolado com a parte do documento onde este está inserido e, então, indicar qual o código ASCII deste. Poderão ser utilizadas algumas outras formas de codificação, como por exemplo o ASCII Estendido ou o BRASCII. O sistema pode também receber os rótulos a serem associados aos caracteres a partir de um arquivo texto, mas deve se ter um cuidado especial para que haja um casamento do rótulo com o padrão adequado.

Um caso particular a ser tratado são os caracteres que se tocam (*touching characters*), onde, conforme já foi descrito, estes serão tratados como apenas um único padrão, mas cujo rótulo é composto pelos códigos de ambos os caracteres. Desta forma, dois caracteres que não puderam ser isolados serão reconhecidos como um elemento único, mas onde o seu código de reconhecimento é um rótulo de mais de um caractere.

Assim, na fase de aprendizado a rede neural recebe caracteres com o seu respectivo rótulo de identificação, possibilitando o aprendizado. Para a fase de reconhecimento, utiliza-se os padrões que receberam automaticamente um rótulo indicando justamente que este é um padrão desconhecido. A rede neural, ao receber um padrão com o rótulo de desconhecido, então, realizará o reconhecimento, devolvendo o código referente ao padrão apresentado.

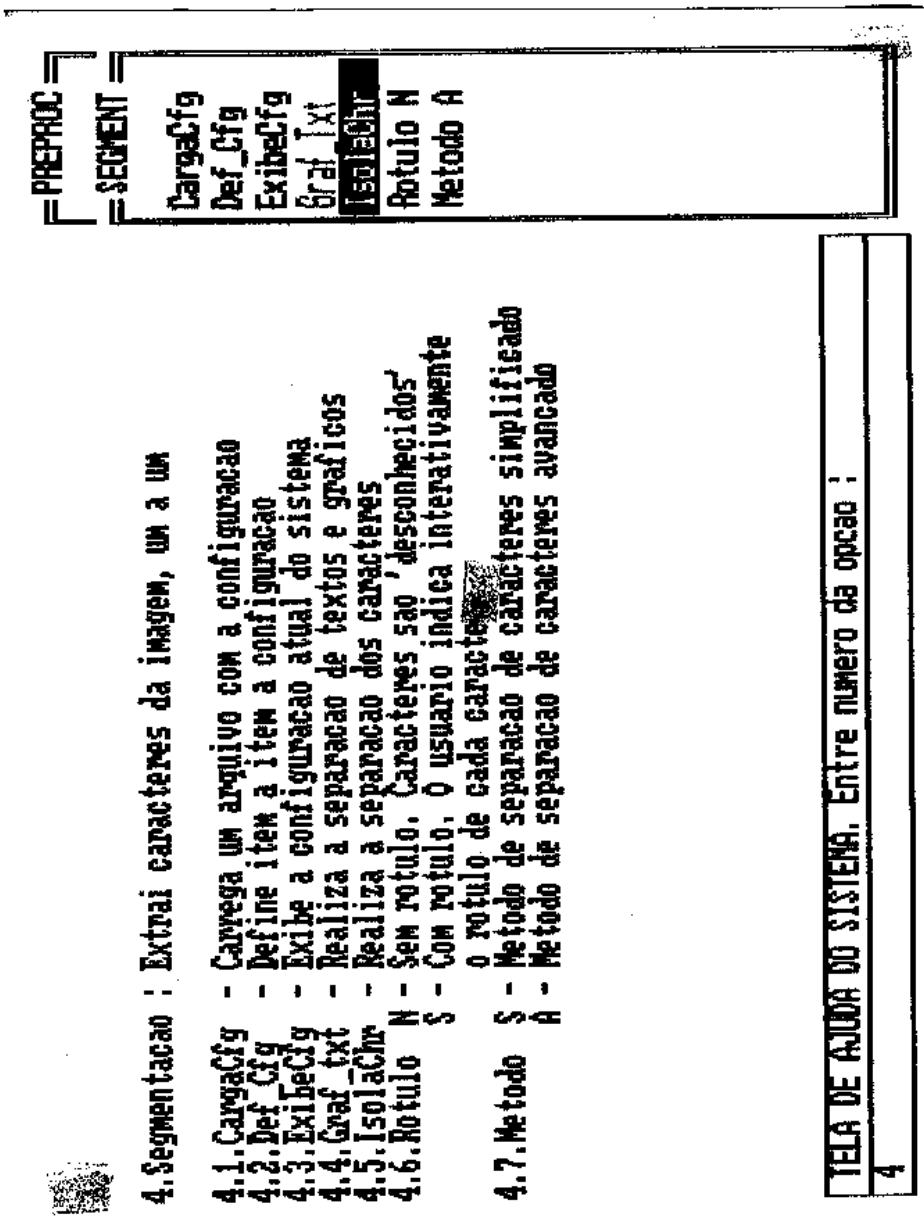


Figura 5.24 - Configuração do processo de rotulação

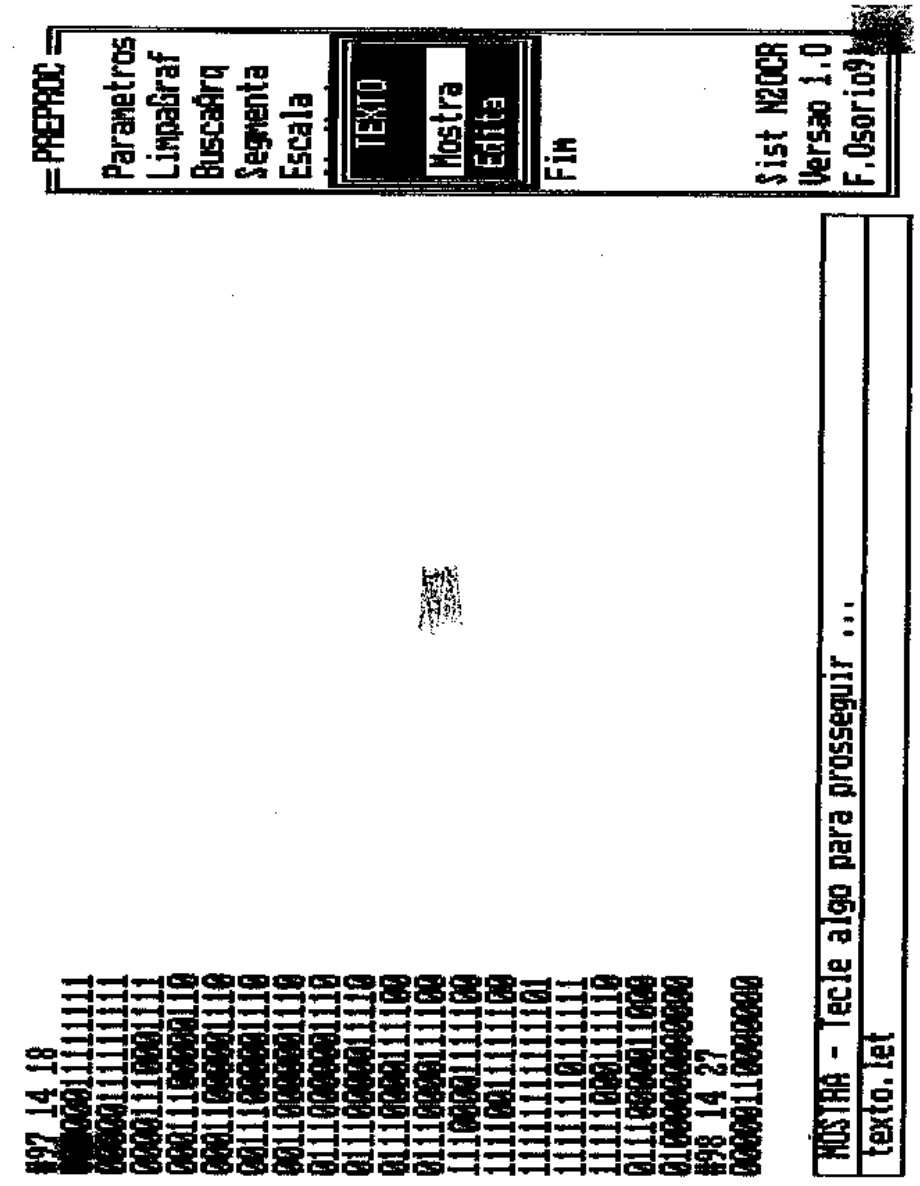


Figura 5.28 - Caractere isolado com código de identificação

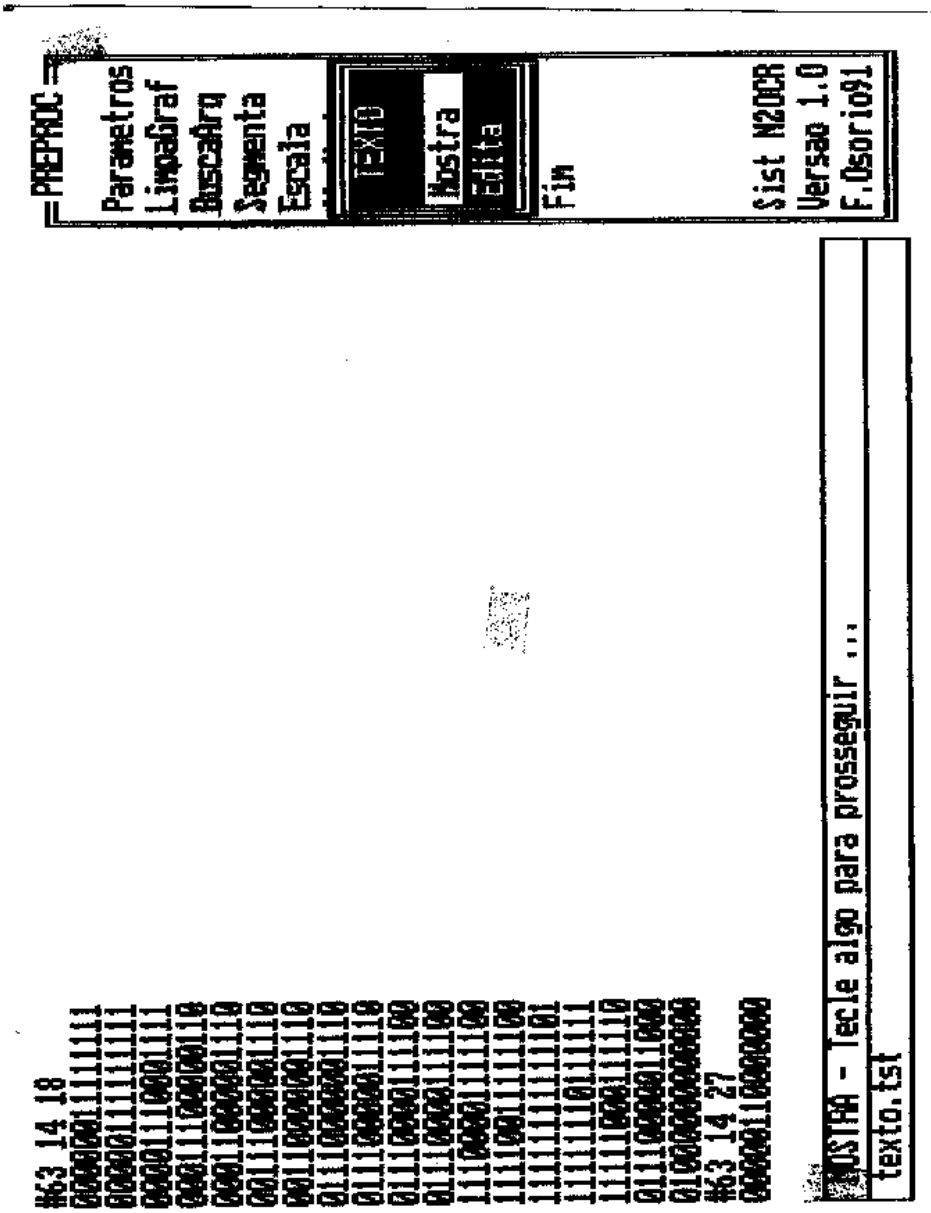


Figura 5.26 - Caractere isolado com rótulo de desconhecido

A figura 5.25 apresenta um caractere isolado com o seu respectivo código de identificação ('a' = ASCII 97), e a figura 5.26 apresenta um caractere com o rótulo que identifica que este caractere é desconhecido ('?' = ASCII 63). O código de identificação de caracteres desconhecidos é um parâmetro configurável do sistema.

5.4.3 Ajuste de Escala e Posição

A etapa final da preparação dos dados para a rede neural é o ajuste de escala e posição, que foi implementado seguindo exatamente a descrição feita na seção 3.4.3. Este algoritmo é de fundamental importância no processo de reconhecimento, porque a rede neural necessita de uma entrada com um número fixo de elementos, para poder efetuar o reconhecimento. Devido a este fato, o ajuste de escala é indispensável, uma vez que permite a padronização do tamanho dos caracteres segmentados pelo processo realizado na etapa anterior. O ajuste de posição também possui uma importância destacada, pois o processo de reconhecimento empregado, na versão atual do sistema, baseia-se no casamento direto de padrões. A figura 5.27 apresenta as opções implementadas na função de ajuste de escala e posição, e as figuras 5.28, 5.29 e 5.30 apresentam exemplos de ajuste de escala em diferentes tamanhos.

5.5 Rede Neural

O módulo de reconhecimento é implementado através de uma rede neural. Esta rede neural adota um modelo específico, criado especialmente para esta aplicação e denominado de: ADAn-LIeNE - *ADAptive Non-LInear Expandible NEtwork*. Esta seção descreverá a estrutura, o funcionamento e a forma de atuação do modelo proposto para atuar junto ao sistema N^2OCR.

Figura 5.27 - Ajuste de escala e posição

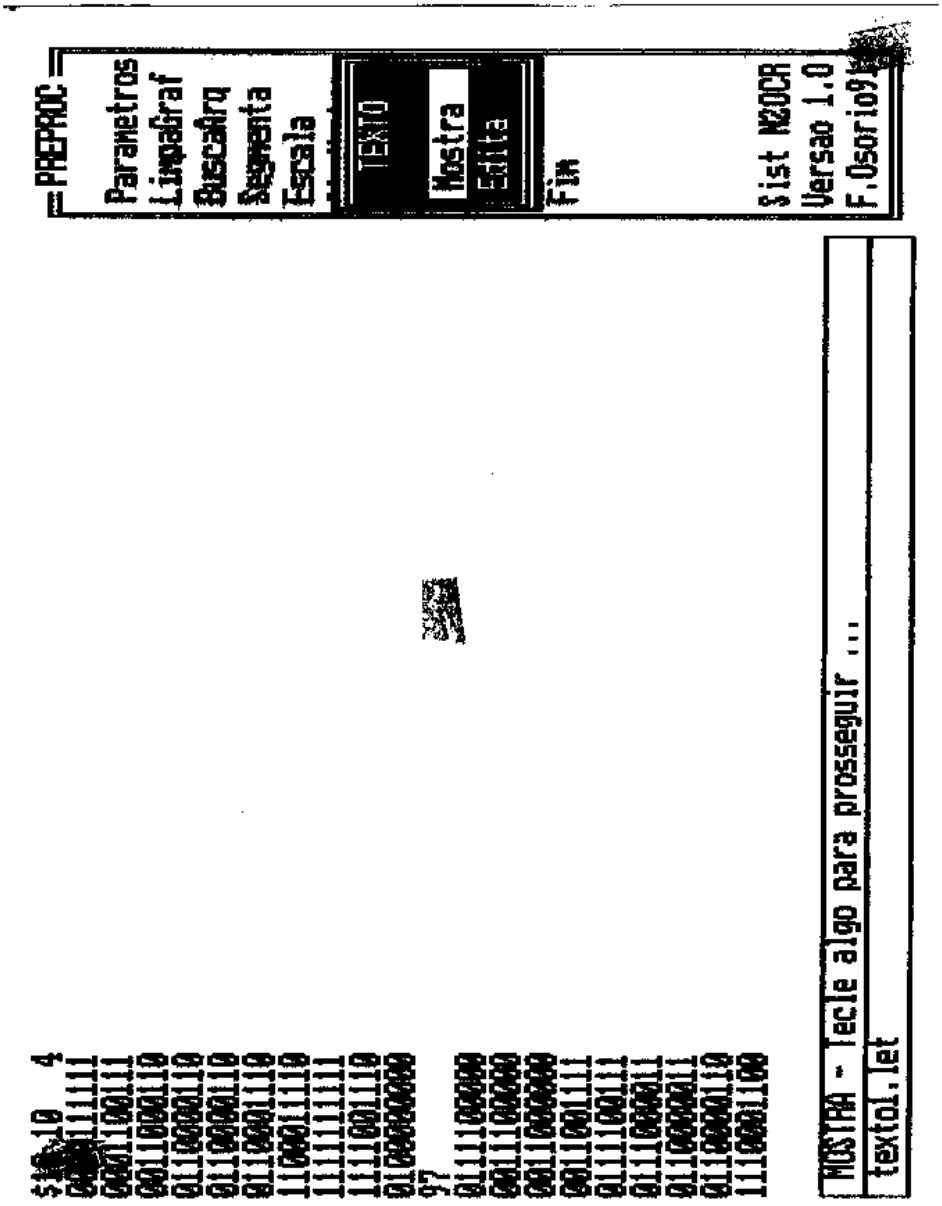


Figura 5.28 - Padrão com ajuste de escala 10 x 10

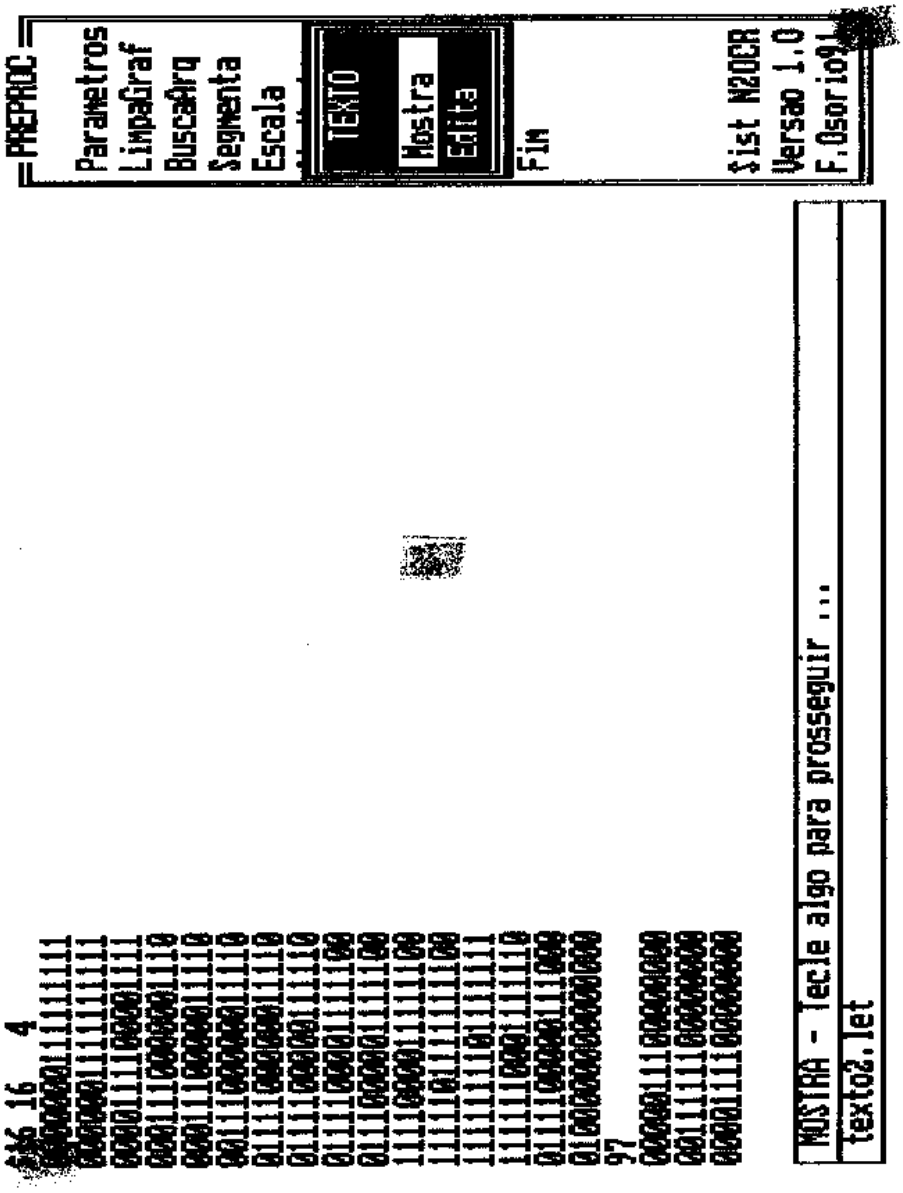


Figura 5.29 - Padrão com ajuste de escala 16 x 16

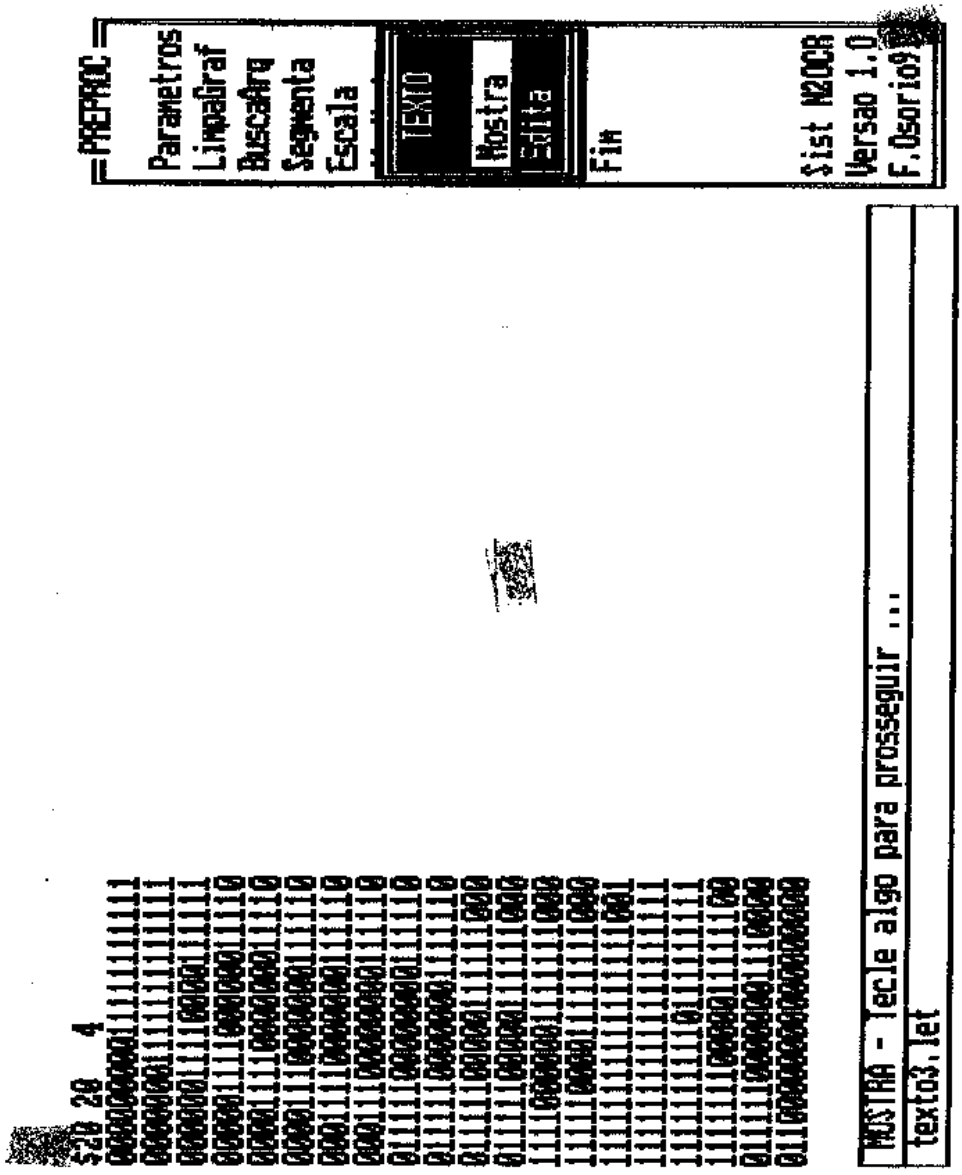


Figura 5.30 - Padrão com ajuste de escala 20 x 20

5.5.1 Modelo Adotado

O modelo de rede neural adotado - ADAn-LIeNE, como o próprio nome já indica, é baseado no ADALINE de Widrow e Hoff. Este modelo resultou da análise dos diferentes modelos de redes neurais adotados na atualidade, onde se buscava um modelo específico para um sistema OCR. O modelo adotado deveria suprir as necessidades de tais aplicações, de maneira simples e eficiente.

5.5.1.1 Escolha do Modelo

Será apresentada aqui uma rápida análise dos modelos já apresentados no capítulo 4, justificando a escolha e desenvolvimento do modelo ADAn-LIeNE.

Os modelos competitivos como o modelo de Hopfield e outros similares, como a máquina de Boltzmann e a BAM, foram os primeiros a serem eliminados da lista de possíveis modelos a serem adotados. Tais modelos possuem alguns inconvenientes como: o tamanho excessivamente grande das redes, a convergência para estados espúrios e a dificuldade de implementação direta em *hardware*. Analisando o modelo de Hopfield, como sendo o representante desta classe de modelos, nota-se que para este modelo ainda seria necessário uma complementação da rede neural, uma vez que esta apenas "corrige" as distorções, devolvendo o padrão restaurado. A identificação do padrão ainda tem que ser feita numa etapa posterior. Outro problema é referente à própria estrutura da rede, onde o seu tamanho é demasiadamente grande e o número de conexões, que interligam todos os neurônios entre si, praticamente inviabiliza uma implementação em *hardware*, onde cada neurônio seria considerado como um elemento individual de processamento. Como um dos objetivos finais deste trabalho é justamente a proposta de um modelo de rede neural que possa ser implementado em *hardware*, estes modelos foram descartados.

Apesar do modelo de Kohonen ser um dos modelos mais interessantes, pela forma similar à realidade com que este representa o comportamento do cérebro humano, este modelo também foi descartado. O principal motivo foi o fato deste modelo gerar os mapas de atributos através da criação de zonas de ativação na rede neural, mas para identificar estas zonas de ativação seria necessário um novo sistema de reconhecimento de padrões. A complexidade de tal operação não justificava a utilização de tal modelo.

Restaram os modelos adaptativos por correção de erros, que somente pela sua capacidade de adaptação já satisfazem a uma das principais necessidades de implementação de sistemas OCR. Dentro destas classe de modelos, há um grupo com aprendizado supervisionado e um modelo de auto-organização, o ART.

Os sistemas de OCR possuem a característica de seu aprendizado ser realizado através de um processo supervisionado. Isto é devido ao fato de que cada novo padrão a ser ensinado ao sistema já possui uma identificação prévia, que é fornecida ao sistema. De nada adianta um sistema onde haja uma auto-organização, e onde os padrões são agrupados (*clusterization*) de acordo com suas características, pois este agrupamento podem não se dar da forma desejada. Um algoritmo de agrupamento pode vir a colocar em uma mesma classe os caracteres "8" e "B", por exemplo, apesar destes não pertencerem a uma mesma classe. Em resumo, em um sistema de OCR, as classes já são bem definidas (conjunto de caracteres), o que implica no uso de algoritmos com aprendizado supervisionado.

Isto quer dizer que o modelo ART deve ser completamente desconsiderado? Não, existe uma propriedade muito importante no modelo ART que será adotada no modelo do ADAn-LIeNE. Esta propriedade é a da expansão da rede, do aumento do número de *clusters* no modelo ART. Cada vez que a rede é apresentada a um novo padrão que não se enquadra, dentro de certos limites, aos padrões já conhecidos, a rede

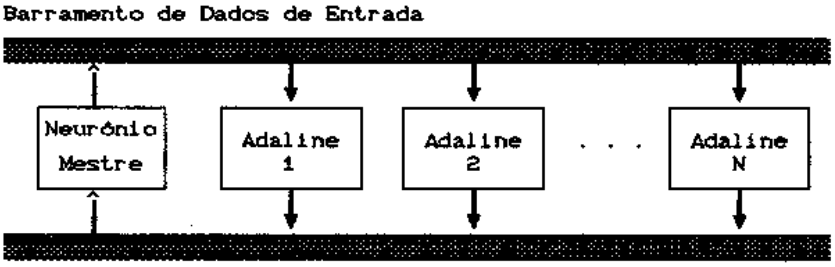
cria um novo nodo responsável pelo aprendizado deste novo padrão. Conforme será apresentado a seguir, o modelo do ADAn-LIeNE também possuirá esta capacidade de expansão, que torná-lo-á mais eficiente.

Do modelo ART foi extraída a capacidade de expansão da rede, durante o processo de aprendizado e adaptação a novos padrões. E do modelo do Adaline foram extraídas as características dos neurônios utilizados na rede. O modelo ADAn-LIeNE utiliza neurônios exatamente iguais aos propostos no modelo do Adaline, com aprendizado através da Regra Delta. Este algoritmo de aprendizado garante à rede uma capacidade de adaptação muito visada em sistemas OCR. A rede neural que se utiliza de neurônios do tipo Adaline possuirá a capacidade de se ajustar aos padrões.

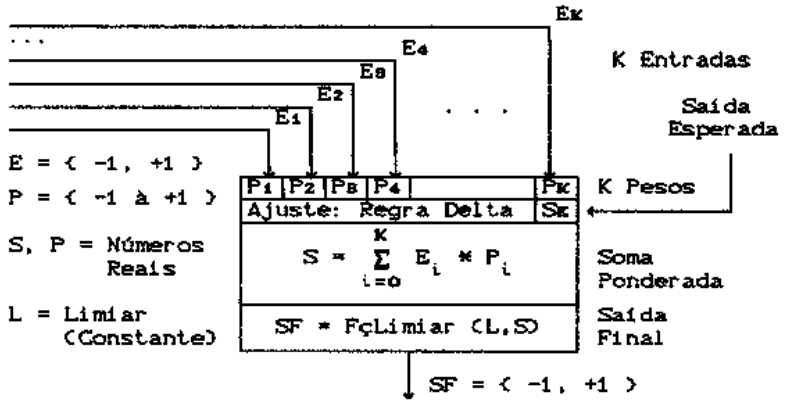
E quanto ao problema da linearidade das decisões tomadas pelo Adaline (rede de apenas uma camada), que levavam a impossibilidade de reconhecer padrões como o XOR? É exatamente devido a este fato que a rede implementada faz uso de uma abordagem de expansão tipo ART, usando também os conceitos definidos por Widrow na criação da rede Madaline. Desta forma as redes são dotadas de decisões não lineares. É justamente destas características que surge o nome adotado para este modelo: *Adaptive Non-Linear Expandible Network*.

Resta apenas a análise de um último modelo, o modelo multinível baseado na regra de *back-propagation*. Este modelo poderia ter sido adotado, sendo inclusive um modelo muito utilizado em diferentes aplicações das redes neurais na atualidade. Mas como a sua grande vantagem está em implementar uma rede neural com decisões não lineares, e isto já foi obtido através do emprego de uma rede similar ao modelo Madaline, torna-se desnecessária a sua utilização. Outros motivos de não ter sido empregado o modelo multinível foram: a sua complexidade, a dificuldade de implementação deste modelo diretamente em *hardware* e o problema da estimativa do número de neurônios e camadas necessárias para uma dada aplicação.

O modelo do ADAn-LIeNE, conforme será demonstrado logo a seguir, é extremamente simples e possibilita uma fácil implementação em *hardware*.



(a) Visão Global da Rede Neural



(b) Elemento Processador Individual - Neurônio Adaline

Figura 5.31 - Organização da rede neural do modelo ADAn-LIeNE

5.5.1.2 Descrição do Modelo

O modelo ADAn-LIeNE é composto por um conjunto de neurônios do tipo Adaline, onde cada neurônio recebe como entrada todos os elementos que constituem um determinado padrão. Cada um dos neurônios será responsável pelo aprendizado de uma determinada classe, ou seja, cada neurônio tem por função o reconhecimento de um caractere específico. A figura 5.31 apresenta a organização da rede neural do modelo ADAn-LIeNE. A rede neural deste modelo, conforme pode ser visto na figura 5.31, possuirá um elemento especial, o neurônio mestre (*Master Neuron*), sendo este responsável pela coordenação do aprendizado e reconhecimento na rede neural, bem como do controle da expansão desta.

A rede proposta se utiliza apenas de valores binários de entrada, representando os elementos de uma imagem de apenas dois níveis. Todos os modelos de redes apresentados e analisados eram orientados ou permitiam a sua utilização para este fim específico. É importante salientar que existem outros modelos de redes neurais que aceitam como entrada valores com mais de dois estados, mas estes modelos, voltados ao processamento de imagens em geral, não são de interesse neste trabalho e não foram objeto de estudo, apesar de sua destacada importância.

O modelo proposto permite que os padrões binários, representando os caracteres, sejam apresentados à rede de forma que apenas um neurônio se ative identificando o caracter. Para que isto seja possível, é necessária a realização de uma fase de aprendizado, para somente após realizar o reconhecimento.

5.5.2 Aprendizado

A fase de aprendizado é realizada recebendo-se, um a um, os padrões juntamente com o código ASCII associado a estes. O neurônio mestre é quem gerenciará estes padrões, determinando qual o neurônio que será treinado para reconhecer cada um dos padrões.

Inicialmente a rede começa a operar, na fase de aprendizado, com todos os neurônios inativos, exceto o neurônio mestre que coordena todo o funcionamento da rede. O neurônio mestre seleciona então o primeiro padrão e "ensina" este a um primeiro neurônio selecionado. O mestre, então, cria uma tabela de controle de aprendizado, onde será descrito o número de cada neurônio em uso na rede (cada um é identificado por um número ou endereço único) e o código do padrão (ASCII) ensinado a este. Para cada novo padrão a ser ensinado à rede, o mestre consulta esta tabela para ver se o código do padrão já está definido ou não. Caso o código do novo padrão a ser aprendido não esteja na tabela, o mestre seleciona um novo neurônio, ensinando a este o novo padrão, e adicionando mais uma entrada na tabela de controle. Caso o código deste novo padrão já exista na tabela, o mestre apenas testa a rede, verificando se o neurônio que foi treinado para identificar este caractere se comportará da maneira correta, ativando-se. Pode ser necessária uma adaptação (ajuste dos pesos) no caso de se ter no aprendizado dois ou mais padrões diferentes representando uma mesma classe (caractere).

O aprendizado consiste em enviar o padrão de treinamento a cada neurônio em uso na rede, juntamente com a resposta que este deve fornecer em sua saída (saída esperada - *mentor input*). Ao serem ativadas as entradas da rede, apenas um neurônio deve ter sua saída ativada (+1), isto implica que apenas um dos neurônios deve ter ativada (+1) sua entrada mentora. Através da aplicação da Regra Delta, os neurônios que não obtiverem em suas saídas o valor de ativação correto ajustam os seus pesos de atuação de forma a obter a resposta desejada. Assim, ao ser ativada a rede pelo neurônio mestre, esta se adapta de forma a responder conforme o indicado por ele, ajustando seus pesos pela Regra Delta.

Cada novo caractere a ser ensinado à rede resulta na adição de um novo neurônio, conforme demonstra a figura 5.32. Assim a rede vai se expandindo à medida que novos padrões lhe são ensinados, colocando em uso neurônios que estavam inativos. De uma certa forma, isto equivale a criar novos neurônios na rede à medida que estes são necessários. Este tipo de expansão da rede foi baseado no princípio de funcionamento do modelo ART.

Devido a estas características do modelo ADAn-LIeNE, fica fácil a adição ou remoção de padrões a serem reconhecidos pela rede. Para ampliar o número de classes (caracteres) reconhecidas basta que sejam adicionados tantos neurônios quantas são as novas classes adicionadas à rede, treinando-os para reconhecer as novas classes de padrões. E no caso de não ser mais necessário o reconhecimento de uma dada classe, então é necessário apenas "desligar" o neurônio que era responsável pelo seu reconhecimento.

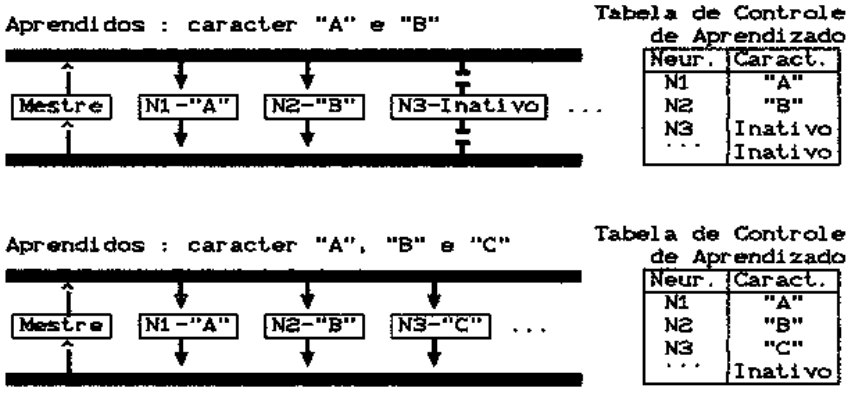


Figura 5.32 - Expansão da rede neural

Ao sofrer o ajuste dos pesos, durante a fase de aprendizado, um neurônio pode "esquecer" o padrão para o qual foi treinado. Quando são alterados os pesos de um

neurônio para este se adaptar a um novo padrão ou para este dar uma determinada resposta, pode ocorrer deste neurônio ter seus pesos modificados de tal forma que não responda mais da maneira correta aos padrões ensinados anteriormente. Devido a este fato, para cada novo padrão ensinado à rede, deve se testar novamente todos os outros padrões ensinados anteriormente, de forma a comprovar que a rede continua a apresentar um comportamento correto diante de cada um dos padrões ensinados a ela. Logo o aprendizado segue a seguinte ordem :

Ensina	Padrão 1	Padrão 2	Padrão 3	Padrão 4
Testa	Padrão 1	Padrão 2,1	Padrão 3,1,2	Padrão 4,1,2,3

O aprendizado segue o algoritmo que é descrito abaixo:

1. Ativa a rede para o padrão N
2. A resposta foi correta ?
   2.1 Não - Adapta os pesos, volta ao passo 1
   2.2 Sim - Segue para o próximo passo
3. Para K igual a 1 até N incrementado de 1
   3.1 Ativa a rede para o padrão K
   3.2 A resposta foi correta ?
      3.2.1 Não - Adapta os pesos e volta para 3
      3.2.2 Sim - Passa para o próximo valor de K
   3.3 Repete o passo 3.1 até acabar todos os padrões de treinamento (até K ser igual a N)
4. Fim

É importante salientar que a rede só terminará de realizar a adaptação a um novo padrão no momento em que esta for ativada para todos os padrões já ensinados, fornecendo a resposta correta para todos eles. Caso ocorra um erro, corrige-se os pesos e volta-se a testar os padrões a partir do primeiro.

O simulador implementado, deste primeiro modelo, realizava a execução exata do algoritmo descrito acima. O ajuste dos pesos era feito através da Regra Delta. Sabia-se que para o caso do padrão da função XOR este procedimento resultaria na execução de um laço sem saída, onde o passo 3 do algoritmo nunca chegaria a ser completado. No caso da XOR, um mesmo neurônio não conseguiria nunca ajustar os seus pesos de forma a dar a resposta correta para todos os padrões de entrada. Este fato é devido ao modelo implementado realizar apenas uma função de decisão linear. Mesmo ciente deste fato, acreditava-se que um sistema de reconhecimento de caracteres não deveria nunca ficar sujeito ao aprendizado de um padrão como o da função XOR. Esta linha de pensamento mostrou-se incorreta com o decorrer dos testes para com a rede neural implementada.

Ao ser realizada uma simulação da rede neural para um dado conjunto de caracteres específicos, ocorreu da rede entrar em um laço infinito, durante a fase de aprendizado. Após uma análise mais detalhada do problema, chegou-se à conclusão que este só poderia ser resultante do aprendizado de padrões que não podiam ser aprendidos com a utilização de uma função de decisão linear: os padrões não eram linearmente separáveis...

Foi então que decidiu-se pela implementação de uma nova técnica, a qual, de certa forma, segue os conceitos propostos por Widrow no modelo do Madaline. A técnica adotada consiste do seguinte: ao ser detectado um laço infinito, caracterizado pela alternância de ajuste de pesos em dois neurônios específicos que nunca conseguem chegar ao resultado desejado, o neurônio mestre pára o processo de aprendizado e cria um novo neurônio na rede. Este novo neurônio especializar-se-á em reconhecer o último padrão que estava sendo ensinado à rede e que causou o problema do laço infinito.

Este caso do laço infinito só ocorre quando um novo padrão, pertencente a uma classe já definida anteriormente, é ensinado à rede. Este novo padrão possui a característica de, juntamente com outro padrão pertencente a mesma classe deste, formar uma situação similar ao do aprendizado da função XOR, ou seja, estes dois padrões pertencentes a uma mesma classe não podem ser classificados corretamente por uma função de decisão linear.

Como o caso descrito acima só ocorre quando é feita a tentativa de ensinar um novo padrão à rede, que pertence a uma classe já definida (o padrão é uma nova amostra de um caractere que já foi ensinado a rede), é necessário apenas que este caso particular seja resolvido. A solução deste problema é simples, como já foi descrito anteriormente. Adiciona-se um novo neurônio que se dedicará a reconhecer exclusivamente este novo padrão causador do problema. Surge então uma nova questão, como fazer para que todos os padrões representantes de uma mesma classe sejam reconhecidos como pertencentes a uma classe única? É simples, na tabela de controle de aprendizado é adicionado um novo campo, onde este campo indica se um determinado neurônio foi treinado para a primeira versão de uma classe ou para alguma versão alternativa desta mesma classe. Esta "versão alternativa" é justamente o padrão problema que gerou a necessidade de criação de um novo neurônio para reconhecer uma classe já ensinada à rede. A tabela de controle de aprendizado terá a organização demonstrada na figura 5.33.

Através do uso da tabela de controle, que permite a identificação de versões diferentes de um mesmo caractere, é possível se implementar uma rede similar ao Madaline. Nesta rede pode se ter, assim como no Madaline, dois neurônios tipo Adaline que identificam dois padrões diferentes pertencentes a uma mesma classe. A saída final da rede é uma função "ou lógico" das saídas dos neurônios associados. Este modelo de simulação permite, então, que se

tenha diferentes versões para uma mesma classe, onde a saída final da rede, gerada pelo neurônio mestre, indica qual a classe reconhecida, independente de ter sido o neurônio X ou Y o qué reconheceu uma das versões desta classe.

Nro. do Neurônio	Classe (Caracter)	Versão
0 0 1	"A"	0
0 0 2	"B"	0
0 0 3	"FT"	0
0 0 4	"A"	1
0 0 5	"C"	0
0 0 6	"D"	0
0 0 7	"B"	1
0 0 8	"F"	0

Figura 5.33 - Tabela de controle de aprendizado

Um fator importante a ser considerado, que é reflexo do modelo de rede proposto, é quanto ao tamanho da rede neural. Como a rede neural permite a expansão através do acréscimo de novos neurônios, uma preocupação natural é para que esta rede não cresça demasiadamente. Obviamente, o número mínimo de neurônios será igual ao total de caracteres diferentes a serem reconhecidos (número total de classes). Quanto ao número máximo de neurônios, este dependerá de alguns fatores relacionados com o processo de aprendizado, estes fatores são:

- O limiar (L) usado na função de transferência;
- O fator de ajuste (β) da Regra Delta;
- O número de padrões exemplares de uma mesma classe e
- A distorção destes padrões, ou seja, o grau de diferença entre os padrões pertencentes a uma mesma classe.

Em relação ao limiar da função de transferência, quanto maior for, mais rígido será o processo de classificação. E quanto mais rígida a classificação, mais difícil será adaptar um mesmo neurônio para reconhecer diferentes versões de uma mesma classe. Como conseqüência deste fato, será criado um número maior de neurônios.

O fator β controla a magnitude da alteração dos pesos através da Regra Delta. Caso o ajuste dos pesos seja realizado com variações muito grandes, ficará difícil a implementação de um "ajuste fino" dos neurônios durante a fase de aprendizado. Portanto, quanto maior o valor de β, mais difícil ficará o ajuste dos pesos, e conseqüentemente a rede crescerá pela adição de diferentes versões para uma mesma classe.

O limiar da função de transferência, assim como o valor do fator de ajuste da Regra Delta, são definidos através do arquivo de configuração da rede neural, descrito posteriormente na seção 5.6. Estes valores são aplicados a todos os neurônios da rede, ou seja, todos os neurônios fazem uso do mesmo valor de limiar da função de transferência e do fator de ajuste da regra de aprendizado. É importante salientar que no caso do limiar da função de transferência, este valor pode ser programado para um certo valor a ser utilizado apenas na fase de aprendizado, com a função de fazer a rede neural se comportar conforme a descrição acima, e, na fase de reconhecimento este valor poderá ser redefinido, passando a atuar como um controle do grau de semelhança mínimo entre o padrão de teste e os padrões aprendidos. O fator de ajuste não terá influência sobre o reconhecimento, uma vez que a rede neural desativa a função de adaptação dos pesos durante esta fase.

Tanto o valor do limiar quanto o do fator β devem levar em consideração o número de representantes de uma mesma classe, assim como o grau de distorção destes padrões, pois todos estes elementos juntos determinarão o comportamento da rede em relação a sua expansão.

Um estudo sobre as relações diretas entre estes componentes torna-se muito interessante e produtivo. Este tipo de análise constitui-se em uma possível extensão dos trabalhos desenvolvidos até o presente.

5.5.3 Reconhecimento

A fase de reconhecimento é simples. Primeiramente é desabilitado o ajuste dos pesos, uma vez que a rede só pode realizar este ajuste se for fornecida a resposta correta junto com os padrões e, no reconhecimento, esta resposta não é conhecida.

O reconhecimento consiste do neurônio mestre fornecer o padrão de teste simultaneamente para todos os neurônios em uso da rede. O neurônio mestre receberá as saídas de todos os outros neurônios, verificando qual o neurônio que se ativou. Caso nenhum neurônio tenha se ativado, a resposta será "não reconheceu" (*not match*). Em caso contrário, o neurônio mestre consultará a tabela de controle de aprendizado para poder indicar a resposta final da rede. O mestre, sabendo qual foi o neurônio que se ativou, pode então obter através da tabela de controle o código ASCII do caractere identificado (o número da versão é desprezado).

E se dois neurônios ativarem suas saídas ao mesmo tempo? Neste caso o mestre selecionará um destes como sendo a resposta correta. O processo de seleção pode ser aleatório ou baseado na ordem em que estes aparecem na tabela de controle. É claro que este tipo de metodologia não é a mais indicada para a solução do problema pois pode levar a erros de reconhecimento. Por isto, na próxima seção, são apresentadas algumas sugestões para a melhoria do modelo descrito.

O elemento responsável pela determinação de uma maior ocorrência de casos como "não reconheceu" ou "múltiplos reconhecimentos" é o limiar da função de transferência aplicado na fase de reconhecimento.

5.5.4 Extensão do Modelo

Conforme foi demonstrado no item anterior, pode ocorrer de mais de um neurônio ativar sua saída na presença de um determinado padrão de entrada. Isto pode constituir tanto um problema como também um auxílio no funcionamento do sistema OCR.

No capítulo 3, foi citada a possibilidade de utilização de métodos para a correção do texto reconhecido através do uso de um pós-processamento baseado no contexto. Neste mesmo capítulo foi indicado que o sistema N^2OCR, apesar de não implementar tal função, já possuia prevista a extensão do mesmo de forma a possibilitar este tipo de operação.

Existem basicamente duas possibilidades de pós-processamento para a correção dos textos reconhecidos através do sistema N^2OCR. A primeira é baseada no modelo que foi proposto anteriormente, sem nenhuma alteração. Como neste modelo é possível a rede ativar a saída de mais de um neurônio, para um dado padrão de entrada, basta que o neurônio mestre, ao invés de "sortear" um dos neurônios como sendo o da resposta correta, passe ao módulo de pós-processamento todos os valores obtidos como possíveis respostas do reconhecimento realizado pela rede. O módulo de pós-processamento terá então mais condições para determinar, entre um conjunto de dados possíveis, qual resposta que possui a maior probabilidade de ser o caractere correto, dentro do contexto em que este está inserido.

Através da alteração do valor do limiar da função de transferência, é possível aumentar ou diminuir a incidência de respostas simultâneas de reconhecimento de uma classe pela rede. Um valor elevado para o limiar da função de transferência implica um maior número de respostas tipo "não reconheceu", e um valor mais baixo para este limiar implica um maior número de saídas dos neurônios ativadas ao mesmo tempo.

A segunda possibilidade de utilização de um pós-processamento junto ao sistema N^2OCR requer uma modificação do modelo de rede neural proposto. A modificação ocorre apenas na conexão da saída dos neurônios com o neurônio mestre. Ao invés de enviar apenas valores +1 ou -1 para o mestre, cada neurônio pode passar a enviar diretemente o valor correspondente ao resultado da soma ponderada, sem a aplicação da função de transferência, ou, com a aplicação uma função de transferência do tipo "lógica limiar" ou "sigmóide". Esta alteração no modelo proposto faz com que a saída dos neurônios não forneça apenas valores binários, mas sim uma faixa de valores, a qual deve ser normalizada (valores entre -1 e +1) para fins de simplificação dos algoritmos, permitindo inclusive um melhor acompanhamento do funcionamento da rede neural.

Para determinar qual o neurônio que melhor identificou o padrão, o mestre achará o maior valor de ativação de saída entre os valores recebidos. O neurônio que enviou o maior valor de saída será considerado como aquele que mais provavelmente identificou corretamente o padrão presente em suas entradas. A determinação do maior valor é feita de uma maneira imediata, não afetando a eficiência do sistema. Uma vez que o mestre tem que ler um a um os valores de saída dos neurônios, a comparação pode ser feita imediatamente entre o maior valor lido até o momento e o novo valor recebido, guardando sempre o maior deles.

Nesta segunda proposta, a correção através do pós-processamento é realizada da seguinte forma: o neurônio mestre agora tem a possibilidade de gerar não apenas o código de um único caractere reconhecido, mas também a lista dos *N* melhores candidatos ao reconhecimento, classificados em ordem ascendente de similaridade. É necessário apenas definir o número de elementos que constituirão esta lista de candidatos (adotou-se três) e enviá-los ao módulo de pós-processamento. Este módulo então avaliará entre os candidatos qual o melhor, de acordo com o contexto em que estão inseridos.

Com a eliminação da aplicação da função de transferência, sempre existirá um candidado ao reconhecimento, não ocorrendo a situação onde a resposta é "não reconheceu". A resposta "não reconheceu" poderá voltar a ser obtida adicionando-se um controle, como o realizado pela função de transferência dos neurônios tipo Adaline, ao neurônio mestre. Isto equivale a postergar a aplicação da função de transferência, deixando esta tarefa para o neurônio mestre. A vantagem desta abordagem é o fato do neurônio mestre ter condições de determinar entre dois candidatos qual o que obteve o maior valor de ativação, o que não era possível quando a aplicação da função de transferência ocorria a um nível interno nos neurônios.

Outro ponto interessante de ser ressaltado é o efeito destas alterações realizadas a nível do comportamento da rede neural no processo de aprendizado. O problema surge na aplicação da regra delta, uma vez que modificamos a saída dos neurônios e não se tem mais o conceito de saída igual a +1 significando o reconhecimento do padrão e saída igual a -1 significando o não reconhecimento. Como saber se devemos ou não realizar a correção dos pesos de atuação de um neurônio? Qual é a saída desejada para cada neurônio?

Existem duas soluções principais, a primeira é a de modificar o comportamento dos neurônios, desativando a aplicação da função de transferência, apenas na fase de reconhecimento, mantendo-a da maneira como vinha sendo utilizada durante a fase de aprendizado. A segunda solução consiste em transferir para o neurônio mestre a decisão de realização ou não de uma correção dos pesos da rede. Neste tipo de procedimento, durante a fase de aprendizado, o neurônio mestre irá receber os valores de ativação de cada neurônio, determinando aquele que obteve o maior grau de reconhecimento, ou seja, o maior valor de saída. Se o neurônio que obteve a maior saída é o que deveria identificar o padrão, não precisamos adaptar os pesos.

No outro caso, em que o neurônio que obteve a maior saída não é aquele que deveria reconhecer o caractere, o neurônio mestre ordena que seja aplicada a Regra Delta, no neurônio que teve a maior saída, mas que não deveria ter reconhecido o padrão (reconhecimento incorreto - saída desejada = -1), e no neurônio que não teve a maior saída, mas que era aquele que deveria reconhecer o padrão (erro de não reconhecimento - saída desejada = +1).

Estes diferentes modelos e variações foram todos desenvolvidos através da implementação de um simulador, onde tem-se a possibilidade de configurar uma série de parâmetros, fazendo com que este atue de uma ou de outra forma. O simulador implementado permite a configuração do tipo da função de transferência utilizada na fase de aprendizado, do tipo desta função utilizada na fase de reconhecimento, dos valores de limiar para estas funções e inclusive o fator de ajuste da regra de aprendizado que pode ser diferenciado para o caso de reconhecimento incorreto (fator β de erro) e para o caso de erro de não reconhecimento (fator β de acerto).

Todos estes parâmetros constituem uma série muito ampla de possibilidades de investigação quanto ao desempenho da rede neural junto ao reconhecimento de caracteres. Os parâmetros adotados, tipos de funções de transferência e exemplos de configurações possíveis para o simulador do modelo ADAn-LIene apresentam-se listados no Anexo A.5, sendo também especificados junto a cada gráfico de avaliação do desempenho do sistema (Anexo A.4), qual a configuração empregada. O autor utilizou-se de um conjunto de parâmetros nos testes, os quais resultaram em um desempenho muito bom da rede neural no reconhecimento de caracteres, mas a estimativa de quais são os valores ótimos empregados nestas funções constitui-se de um trabalho extenso e novo, o qual ainda não foi realizado.

5.5.5 Implementação em Hardware

Nesta seção, será apresentada apenas uma visão superficial de alguns aspectos relevantes para a especificação de uma arquitetura especial que implemente o modelo de rede neural descrito neste trabalho. O objetivo é dar ao leitor uma visão geral da possibilidade de implementação em *hardware* do modelo que foi simulado por *software*.

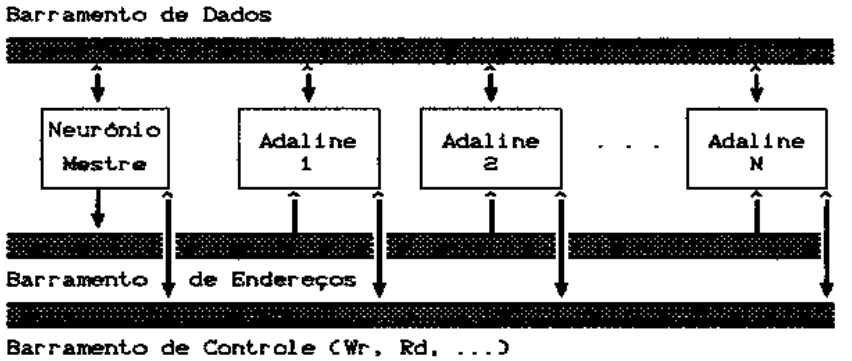
A arquitetura básica do sistema é baseada no esquema apresentado na figura 5.31. O sistema é composto por um conjunto homogêneo de processadores que operam em paralelo e realizam individualmente as tarefas relativas aos neurônios tipo Adaline. Estes processadores estão interconectados por um barramento a um processador mestre. O processador mestre é um processador mais sofisticado (um microprocessador, por exemplo) que executa tarefas de controle da rede neural.

Os dados são eviados através do barramento de dados, onde utiliza-se uma técnica de difusão simultânea (*broadcast*) das informações para todos os elementos processadores (neurônios). Os dados podem ser enviados de uma só vez, caso o barramento possua uma largura que suporte todo o padrão (em torno de 256 bits - 16x16), ou o padrão pode ser particionado enviando-se, por exemplo, de 16 em 16 bits até compor todo o padrão.

O neurônio mestre terá que possuir alguma forma de seleção individual dos neurônios, seja para selecionar o neurônio que enviará seu valor de saída, seja para selecionar qual o neurônio que receberá o valor de saída desejado (*mentor input*). Uma possibilidade de implementação é através do uso de barramento de endereços, onde cada neurônio recebe um endereço específico que será utilizado para o seu acesso. Através desta técnica, o envio do valor de saída desejado pode ser feito selecionando o endereço de um certo neurônio e enviando este dado.

Para a leitura do valor de saída de cada neurônio, o mestre pode realizar um processo de *pooling*, fazendo o acesso a cada um dos neurônios seqüencialmente.

A figura 5.34 apresenta um esquema simplificado de uma arquitetura para implementação do modelo analisado neste trabalho, juntamente com as funções que este sistema deve oferecer.



Funções do Mestre

Envia Padrão ⟶
- Dado : Padrão (*bit-map*) – Mestre envia
- Endereço: Endereço do neurônio que deve se ativar (único com saída = +1)
  - Se endereço = 0 ⟶ Reconhece
  - Se endereço <> 0 ⟶ Aprende
- Controle: Sinal de escrita (Wr)

Recebe Saídas ⟶
- Dado : Valor de ativação - Mestre recebe
- Endereço: End. do neurônio a ser acessado
- Controle: Sinal de leitura (Rd)

Outras funções:
- Inicializa – Pesos, Limiar, Fator β
- Carrega Pesos – Valores pré-gravados
- Lê Pesos – Grava os pesos atuais

Figura 5.34 - Proposta simplificada de uma arquitetura

Uma rede neural pode ser implementada em *hardware*, através do projeto de uma arquitetura dedicada como a descrita acima, ou pode ser desenvolvido um projeto visando o aproveitamento apenas do paralelismo inerente ao processo, mas utilizando-se de uma arquitetura para o processamento paralelo, que não seja necessariamente dedicada apenas a implementação de uma rede neural.

O paralelismo nas redes neurais pode ser explorado de diferentes formas. A primeira, que é a mais clara destas formas, é obtida através de uma implementação onde cada neurônio é um elemento individual de processamento. Este enfoque foi o apresentado através do esquema da figura 5.34. Uma outra possibilidade é o aproveitamento apenas do paralelismo da multiplicação do vetor de entradas pelo vetor de pesos, onde esta multiplicação pode ser realizada em processadores vetoriais, acelerando o processo. Neste enfoque já não se tem mais o conceito de neurônio como sendo um elemento individual. O aproveitamento do paralelismo existente na multiplicação das entradas e pesos pode ser realizado de diferentes maneiras. Um exemplo de implementação de simuladores de redes neurais em arquiteturas paralelas não dedicadas é descrito em [OSO 90b].

5.6 Arquivo de Configuração

Para que o sistema N^2OCR possa se tornar um sistema realmente prático no reconhecimento de caracteres, é necessária a possibilidade de configuração de alguns parâmetros. A configuração do sistema permite que este se adapte de uma forma mais perfeita ao tipo de texto e necessidades de uma dada aplicação do usuário.

A implementação desta configuração do sistema é feita através da utilização de um arquivo de configuração. O arquivo de configuração descreverá os diversos itens referentes ao sistema N^2OCR que podem ser definidos, programados ou selecionados pelo usuário.

Este arquivo de configuração pode ser alterado livremente pelo usuário (arquivo em formato texto) e deve ser consultado pelo sistema na sua inicialização ou quando for solicitada uma reconfiguração do sistema.

O protótipo do sistema N^2OCR não possui ainda implementadas todas as funções de configuração descritas nesta seção. Como nem todos os algoritmos previstos para o sistema N^2OCR estão presentes no protótipo, alguns parâmetros de configuração não foram considerados. As funções implementadas no protótipo, em alguns casos, foram simplificadas e não possuem todas as possibilidades de configuração que poderiam fazer uso. Outro detalhe a respeito da implementação realizada no protótipo é o fato de cada módulo possuir a sua própria configuração, sendo que no sistema N^2OCR está prevista a utilização de um arquivo único para a configuração geral de todas as ferramentas.

Uma função muito importante dos arquivos de configuração é a definição de parâmetros para o processamento automático do sistema, com o mínimo de intervenções realizadas pelo usuário. Este tipo de processamento, na implementação disponível atualmente, não é um procedimento comum, onde a configuração e processamento no sistema se dá através da configuração e seleção de opções através de uma interface amigável orientada a menus.

Os parâmetros configuráveis do sistema N^2OCR são os seguintes:

1. *Parâmetros para aquisição de imagens:*

- Definição da fonte da imagem digitalizada, indicando se é obtida diretamente do dispositivo de aquisição de imagens, ou se esta deve ser carregada de um arquivo específico;
- Nome do arquivo que contêm a imagem e
- Formato do arquivo de imagens, podendo ser do tipo CUT, TIF, PCX, BTM ou IMF.

2. *Parâmetros para tratamento das imagens*:

- Tipo de tratamento da imagem, indicando o tipo de filtragem a ser realizada sobre esta;

- Indicação se é ou não necessária a realização de uma operação para desfazer o *halftoning* e

- Indicação se é necessária a realização da binarização da imagem e, no caso da realização desta, fornecimento do valor do limiar utilizado na binarização.

3. *Parâmetros para a preparação dos dados para a rede neural*:

- Indicação do método de separação de textos e gráficos (automática ou interativa);

- Fornecimento dos fatores usados no controle da separação entre textos e gráficos, conforme descrição feita no capítulo 3;

- Tipo de espaçamento dos caracteres, podendo ser espaçamento fixo ou variável (*fixed pitch/proportional*). Indica qual algoritmo de segmentação será empregado;

- Fornecimento dos fatores usados para a determinação de espaços ou linhas em branco na separação dos caracteres entre si, conforme descrição feita no capítulo 3;

- Fornecimento dos fatores para segmentação de caracteres com espaçamento fixo: altura e largura média de um caractere;

- Tipo de detecção inicial dos textos, podendo ser automática ou interativa;

- Indicação dos fatores usados para a detecção automática de textos, conforme descrição do capítulo 3, e

- Indicação da necessidade ou não da extração de atributos, fornecendo qual o processo a ser empregado;

4. *Parâmetros para a simulação*:

- Tipo de código usado na representação dos caracteres: ASCII, EBCDIC, BRASCII, ASCII Estendido ou definível pelo usuário;

- Código de saída da rede para um caractere não reconhecido;

- Especificação do número máximo de neurônios utilizáveis pela rede neural (gera aviso no caso de exceder este número);

- Tipo de procedimento a ser executado: aprendizado ou reconhecimento;

- Se for efetuado um aprendizado, pode ser feita a indicação de que se deseja "salvar" os pesos da rede após realizar este.

- Se for realizado um reconhecimento, pode ser feita a indicação de um arquivo para a carga dos pesos previamente salvos, realizando o reconhecimento baseado em um conhecimento prévio;

- Determinação do comportamento das saídas dos neurônios: saídas com valores binários ou não;

- Definição do tipo de função de transferência empregada na fase de treinamento e de reconhecimento;

- Fixação dos valores do limiar "L" e do fator de ajuste "β" usados pelos neurônios;

- Determinação do limite de retentativas realizadas no aprendizado, antes de realizar uma expansão da rede, e

- Indicação da necessidade de geração ou não de um relatório da simulação. Este relatório descreverá o número total de neurônios utilizados, o número total de ativações da rede, o número de caracteres com mais de uma versão (e quais são estes caracteres), o número médio de ajustes dos pesos por neurônio, etc.

Todos os fatores citados acima permitem um melhor desempenho do sistema frente a diferentes tipos de textos a serem reconhecidos. A obtenção de uma boa taxa de reconhecimento correto de caracteres dependerá da correta configuração e do ajuste mais adequado dos valores de controle do sistema.

5.7 Avaliação do Desempenho do Sistema

Baseando-se nos resultados obtidos a partir da utilização do protótipo do sistema N^2OCR, chegou-se à conclusão que este possui um ótimo desempenho no que se refere ao reconhecimento correto dos padrões. Quando o sistema é treinado para um certo conjunto de padrões e, após, estes mesmos padrões são então reintroduzidos na rede para o reconhecimento, a taxa de reconhecimento correto é sempre de 100%. Esta propriedade é inerente ao sistema, devido ao tipo de modelo adotado, onde o algoritmo de aprendizado só termina a sua execução quando todos os padrões de aprendizado tiverem sido testados e reconhecidos corretamente.

Os testes realizados com o sistema N^2OCR foram realizados com três conjuntos diferentes de padrões de aprendizado/reconhecimento:

- Imagens de caracteres obtidas através da digitalização com um *scanner* manual de textos de uma impressora laser → Denominado de *Padrão de Teste A*;
- Padrões gerados através da utilização de um programa desenvolvido para este fim, com a possibilidade de adicionar distorções → Denominado de *Padrão de Teste B*, e
- Padrões obtidos por um sistema CCD de aquisição de imagens a partir de uma conjunto de dados fornecido pelo correio inglês → Denominado de *Padrão de Teste C*.

Estes três tipos de padrões de teste possuem alguns exemplares que são apresentados no Anexo A.3, onde tem-se alguns padrões típicos de cada amostra que foram empregados na avaliação do sistema N^2OCR. A seguir serão descritos cada um dos padrões de teste utilizados, em relação as características de obtenção dos mesmos (processo de aquisição) e em relação ao tipo de caracteres contidos nestes (fontes, distorção, etc).

O padrão de teste A foi obtido a partir da digitalização com um *scanner* operado manualmente (*handy scanner* HS-3000) produzido pela empresa DFI Inc. A digitalização foi realizada com a obtenção de imagens P&B (2 níveis somente, sem *halftoning*) e com o ajuste de contraste do dispositivo em sua posição média. O *software* de aquisição de imagens utilizado foi o programa SCAN.EXE que acompanha o *scanner*, sendo gravadas as imagens em arquivos do tipo *.CUT. As imagens digitalizadas foram feitas a partir de textos impressos em uma impressora laser de 300 DPI, onde os textos foram editados em uma estação de trabalho SUN, e impressos com uma única fonte, a fonte *default* para impressão na laser. Todo a manipulação posterior das imagens obtidas pelo *scanner* foi realizado utilizando-se o protótipo do sistema N^2OCR.

O padrão de teste B foi produzido com o programa PADRÃO.EXE que é um dos módulos do protótipo do sistema N^2OCR. Os padrões de caracteres obtidos já são gerados de forma isolada, não havendo a necessidade de empregar um algoritmo de segmentação. Os padrões foram gerados com a utilização de uma única fonte, equivalente aos caracteres *default* do TurboC (Compilador C - Borland Inc.) para o modo gráfico. Para efeitos de teste do reconhecimento foram geradas distorções apenas com a inversão de pixels do padrão. Nos testes com o padrão B não foram adicionadas outras distorções como rotação ou deslocamento do padrão.

O padrão de teste C foi fornecido pelo prof. Edson Filho da Universidade Federal do Pernanbuco, sendo esta base de dados utilizada por este professor nos testes realizados em sua tese de doutorado [FIL 90 e FIL 91]. O conjunto de dados é formado por caracteres alfanuméricos impressos extraídos de envelopes postais e fornecidos pelo Correio Britânico. A digitalização foi realizada através de um dispositivo CCD de aquisição de imagens. A base de dados utilizada é composta por 34 classes de caracteres, obtidos a partir de múltiplas fontes (diferentes tipos de

impressão), onde tem-se as letras de A até Z (excluindo-se a letra 'O' e a letra 'I' - 24 classes), e os dígitos de 0 a 9 (onde os números 'O' e o '1' não são distingüidos das letras 'O' e 'I' - 10 classes). Na base de dados, cada classe contém 300 amostras de padrões digitalizados em 384 pixels P&B, organizados em uma matriz de 16 x 24 pontos. Os padrões já se encontram isolados, tendo sofrido um processo anterior de segmentação e ajuste de posição e escala. Para a utilização desta base de dados foi realizada apenas a conversão do formato de arquivo dos padrões para os formatos utilizados pela rede neural (conversão para arquivos do formato tipo *.LET e *.TST), adicionando-se os respectivos rótulos para permitir o aprendizado e reconhecimento.

Os testes realizados sobre o sistema N^2OCR foram concentrados principalmente sobre os padrões de teste A e C, por estes reproduzirem mais a realidade de um sistema OCR, utilizando dados extraídos de casos reais. Os testes e resultados obtidos com os diferentes tipos de padrões são apresentados no Anexo A.4. Estes testes fizeram uso de informações geradas nos arquivos *.REC e *.REL (vide Anexo A.5), sendo também utilizada a função de avaliação implementada no módulo PADRÃO (fig. 5.35).

Os teste com o padrão A obtiveram resultados excelentes, onde mesmo com um pequeno treinamento a rede atinge um alto grau de reconhecimento correto dos caracteres. Nestes testes o grau de reconhecimento correto ficou entre 98 e 100%, demonstrando que para este tipo de aplicação o sistema N^2OCR está plenamente apto a ser utilizado. Outra característica importante deste teste é o fato de envolver todas as etapas de processamento realizadas pelo sistema N^2OCR, da aquisição até o reconhecimento.

Figura 5.36 - Avaliação automática do reconhecimento

Os resultados obtidos nos testes com o padrão C são muito bons e devem ser analisados cuidadosamente, uma vez que este padrão apresenta um alto grau de distorções e onde o texto está escrito com a utilização de múltiplas fontes, o que, mesmo em sistemas profissionais de OCR, reduz em muito a eficácia destes sistemas. O grau de acertos no reconhecimento é considerado bom, sendo que um treinamento mais cuidadoso pode certamente melhorar a eficácia do sistema atingindo níveis como os obtidos nos testes com o padrão A.

Quanto ao reconhecimento de padrões com distorções (padrão de teste C), onde o conjunto de reconhecimento possui diferenças significativas em relação ao conjunto de aprendizado (padrões perfeitos), a rede neural também teve um ótimo comportamento. No caso de padrões distorcidos por inversão aleatória de pixels, os resultados obtidos demonstram uma exatidão de 90 a 100% de acertos quando a rede é submetida a padrões com distorções de 5 a 20% do conjunto total de pixels deste. É claro que todos estes resultados variam conforme ajustes da configuração do simulador da rede. O Anexo A.4 - Padrão B, apresenta a avaliação realizada com este tipo de padrões gerados com uma distorção controlada dos pixels.

O sistema, na fase de reconhecimento, se mostrou pouco imune a problemas como os caracteres transladados ou rotacionados, que elevam consideravelmente as taxas de reconhecimento incorreto ou de não reconhecimento de caracteres. Este comportamento já era esperado dadas as características do sistema, que realiza um casamento direto dos padrões. No entanto, basta a realização de um aprendizado coerente com o tipo de caracteres que serão reconhecidos, e feito para um número de padrões representativo em relação aos diferentes tipos de padrões de cada classe, que o sistema terá a sua eficácia aumentada significativamente.

Uma avaliação ainda mais detalhada do comportamento do sistema, torna-se muito difícil por uma série de fatores, entre estes estão alguns já citados no capítulo 3. Um dos outros fatores que dificultou bastante a avaliação do desempenho do sistema foi o problema referente ao alto tempo consumido na simulação da rede neural, durante a fase de aprendizado. Como o simulador implementado utiliza-se de um processador único (IBM-PC com processamento seqüencial) na sua execução, e onde a rede neural simulada possui um alto grau de paralelismo (que terá que ser simulado de maneira seqüencial) isto implica um alto custo em termos do tempo gasto para o de processamento de informações. Os programas de simulação de redes neurais usualmente gastam muito tempo de processamento, e, no caso do simulador do modelo ADAn-LIeNE, em simulações de complexidade média, eram consumidas em torno de 12 horas de CPU em um equipamento do tipo IBM-PC 386 compatível com co-processador aritmético e operando a 25 Mhz. Este alto tempo gasto é relativo a fase de aprendizado apenas, que é extremamente demorada devido aos ajustes e repetição contínua de ativações da rede até o aprendizado perfeito. A fase de reconhecimento pode ser executada sem problemas de demora excessiva do processamento dos dados. Devido a este fato, não foi possível a realização de testes exaustivos sobre o modelo implementado. Como tem-se um grande número de variáveis envolvidas neste processo, seriam necessárias muitas rodadas de aprendizagem para estabelecer os valores ótimos dos fatores programáveis do sistema. Estes valores ainda tem que ser determinados para cada conjunto diferente de padrões de aprendizado e reconhecimento, onde se obtem comportamentos diferentes da rede.

A avaliação do sistema, segundo o autor, deve ser realizada através do uso de uma base de dados padrão, onde se possa comparar os resultados obtidos com os resultados de outros sistemas OCR, que também foram submetidos aos mesmos testes. Somente assim será possível se ter uma noção mais exata do desempenho real do sistema.

Observando-se o comportamento geral do sistema, foi possível notar algumas de suas características que comprovam a sua versatilidade:

- Capacidade de adaptação: ajuste a novos padrões;
- Aprendizado do reconhecimento de textos com múltiplas fontes de caracteres: o número de caracteres diferentes aprendidos só é limitado pelo tamanho máximo da rede neural;
- Possibilidade de implementação em *hardware* paralelo, obtendo-se uma alta *performance* no reconhecimento;
- A utilização de simuladores permite uma maior flexibilidade no estudo, depuração e ajustes do modelo adotado, antes de sua implementação em *hardware*;
- O modelo empregado ressalta o paralelismo existente no processo de reconhecimento, facilitando a sua implementação em máquinas que permitem o processamento paralelo;
- O "salvamento" dos pesos da rede permite a criação de arquivos com o treinamento (fase crítica em termos de tempo de processamento) já realizado para determinados tipos de textos, e
- Possibilidade de extensão do sistema para adição de pós-processamento, sem maiores alterações deste.

Devido a todas estas características o sistema N^2OCR mostra-se como uma ferramenta de alta flexibilidade, possuindo recursos que tornam este muito adequado ao reconhecimento de caracteres.

5.8 Evolução do Sistema

O sistema N^2OCR foi implementado de maneira que este possa sofrer adaptações e expansões visando a sua melhoria. Um exemplo disto é o caso da possibilidade de adição de um módulo de pós-processamento para a correção do texto reconhecido através do emprego do contexto.

Uma outra possibilidade de expansão está relacionada com a preparação dos dados para a rede neural. Conforme foi apresentado na seção 3.4.2, os padrões utilizados pelos sistemas OCR podem sofrer um pré-processamento, visando a extração de algumas de suas características. O sistema N^2OCR é um sistema que realiza o casamento de padrões, determinando, para um dado padrão desconhecido, qual o padrão de aprendizado que mais se parece com este. A implementação atual do sistema utiliza como padrão de entrada da rede neural o próprio *bit-map* do caractere extraído da imagem digitalizada. O sistema não possui nenhuma restrição que impeça este padrão de entrada de sofrer uma recodificação, visando ressaltar algum tipo de atributo específico ou, até mesmo, o *bit-map* do caractere pode ser estendido pela adição de novas informações complementares.

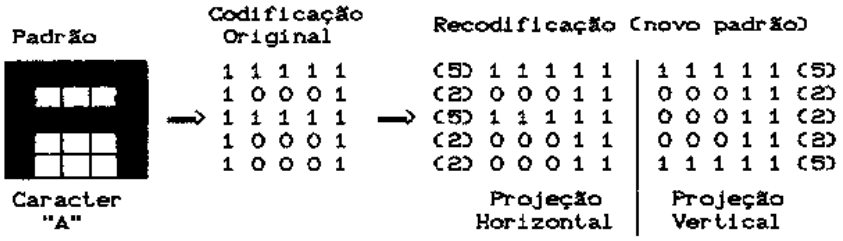


Figura 5.36 - Recodificação de um padrão

A recodificação serve para ressaltar, para a rede neural, algum tipo de característica do *bit-map* do caractere que pode estar "escondido" neste, e não esteja sendo levado em consideração, ou seja pouco considerado, para a obtenção do resultado do reconhecimento efetuado pela rede neural. Através da recodificação dos padrões, pode-se destacar para a rede algum tipo de propriedade específica destes, de forma que esta propriedade seja levada em consideração no reconhecimento. É necessário se ter muito cuidado ao realizar esta recodificação, pois o reconhecimento realizado

pela rede neural mede, de certa forma, apenas a distância do padrão de teste em relação aos padrões de aprendizado, indicando o mais próximo.

Para o exemplo da figura 5.36, é interessante se notar que o valor 5 foi codificado como "11111" e não em binário, que seria igual a "00101". Isto se deve ao fato do valor "00101" em binário ficar muito parecido a outros valores, como "10101" que equivale a 21, onde, apesar da similaridade entre ambos padrões ("00101" e "10101"), estes representam dois valores extremamente diferentes. Este tipo de metodologia, onde aplica-se um pré-processamento dos padrões, tem sido avaliado como sendo uma forma alternativa de funcionamento do sistema N^2OCR, em um futuro próximo. A extensão do sistema N^2OCR através desta técnica possibilitará que este se torne mais insensível a certos tipos de deformação como rotações ou translações dos caracteres (conforme o tipo de recodificação empregado). Desta forma, pretende-se dotar o sistema da capacidade de reconhecimento inclusive de caracteres manuscritos em letras de forma. Para isto é necessário um estudo, avaliação e determinação do melhor método de recodificação dos caracteres.

Além da expansão do sistema, outra tarefa que vem sendo realizada é a tentativa de obtenção de uma base de dados de teste padrão, que possa ser utilizada junto ao N^2OCR visando a melhor validação do modelo implementado. Também tem sido realizados estudos no sentido de portar o sistema para máquinas onde seja possível a utilização de um processamento paralelo, seja com processadores fracamente acoplados (por exemplo, uma rede de *workstations* SUN) ou através de processadores fortemente acoplados (por exemplo, um *hardware* baseado em *Transputers*). Cria-se desta forma um simulador que executa suas tarefas em paralelo, realizando uma parte do processamento junto a cada processador.

O sistema N^2OCR também propiciou, como resultado de seu desenvolvimento, um estudo mais aprofundado das rede neurais. Este estudo resultou na criação de um pro-

jeto denominado SIRENE - Simuladores de Redes Neurais. Este projeto garante a continuidade dos estudos realizados sobre este tema, e visa a implementação de ferramentas para a análise e utilização de modelos de redes neurais de destaque na atualidade, assim como permite investigações com o objetivo de criar novos modelos de redes neurais. O projeto SIRENE está ligado ao grupo de estudos de arquiteturas avançadas de computadores (processamento paralelo), coordenado pelo prof. Philippe O. A. Navaux, junto ao CPGCC da UFRGS. Além deste projeto, que se encontra em pleno desenvolvimento, os estudos realizados para o desenvolvimento deste trabalho também tem sido utilizados para colaborar com o desenvolvimento de um "processador neural", que está sendo feito sob a coordenação do prof. Paulo M. Engel.

Em resumo, o estudo realizado neste trabalho abriu uma série de novas perspectivas de estudo dentro das linhas de pesquisa sobre redes neurais e reconhecimento de caracteres. O sistema N^2OCR permite também uma evolução, sendo possível a realização de novos estudos para complementar o trabalho que já foi desenvolvido, a nível de avaliação do desempenho, estudo de convergência, estimativa de parâmetros ótimos de configuração, pós-processamento do texto reconhecido, etc.

6 Conclusão

Este trabalho teve por objetivo o estudo do reconhecimento visual de caracteres através da utilização das redes neurais. Para atingir tal objetivo foram desenvolvidos os tópicos referentes ao Processamento Digital de Imagens, aos sistemas de reconhecimento de caracteres (OCR), às redes neurais e por fim foi criada uma proposta de solução e implementação de um sistema OCR, o denominado sistema N^2OCR. Para validar esta proposta de implementação foi desenvolvida uma série de ferramentas que implementam as diversas operações e etapas de processamento estudadas neste trabalho, criando assim um protótipo do sistema N^2OCR.

Os resultados obtidos na utilização do protótipo do sistema comprovam que o N^2OCR não é apenas viável, como também demonstram ser este uma ótima alternativa como sistema para reconhecimento de caracteres impressos. Este sistema demonstra uma série de propriedades que tem sido apresentadas como muito desejáveis, senão essenciais, em sistemas OCR.

O trabalho demonstrou-se muito válido também pelos seus reflexos, como foi o caso da criação do projeto SIRENE no CPGCC da UFRGS. Outro aspecto importante foi o fato deste não ter sido um trabalho limitado, pois permite expansões que podem melhorar ainda mais o seu desempenho, chegando até possivelmente ao reconhecimento de caracteres manuscritos.

Além de incentivar o desenvolvimento de novas frentes de estudos junto ao CPGCC da UFRGS, este trabalho também possibilita que se chege a resultados com aplicações práticas, não somente para fins científicos. O sistema desenvolvido pode ser aperfeiçoado, através da complementação de alguns módulos e da implementação em *hardware* de uma arquitetura paralela que realize as funções executadas através do simulador da rede neural, chegando ao nível dos sistemas comerciais de OCR.

É importante salientar que este trabalho, por tratar de uma área pouco estudada em termos de Brasil, e principalmente em relação ao CPGCC da UFRGS, buscou a formação de uma base de conhecimentos sobre o assunto. Desta forma todo o trabalho serve para dar subsídios para novos trabalhos que possam vir a ser desenvolvidos em áreas relacionadas com as que foram abordadas.